

Retrocomputer Magazine Anno 5 - Numero 27 - Gennaio 2010


Start System


Install System Software


Run Diagnostics


Recover System


Enter Command Monitor

W E L C O M E T O Indigo

Silicon Graphics Computer Systems

Jurassic News

Rivista aperiodica di Retro-computing

Coordinatore editoriale
Tullio Nicolussi [Tn]

Redazione
Sonicher [Sn]
redazione@jurassicnews.com

Hanno collaborato a questo numero:
Tullio Nicolussi [Tn]
Salvatore Macomer [Sm]
Lorenzo 2 [L2]
Besdelsec [Bs]
Lorenzo Paolini [Lp]
Mario Raspanti [Mr]

Impaginazione e grafica
Anna [An]

Diffusione
marketing@jurassicnews.com

La rivista viene diffusa in formato PDF via Internet agli utenti registrati sul sito www.jurassicnews.com. la registrazione è gratuita e anonima; si gradisce comunque una registrazione nominativa.

Contatti
info@jurassicnews.com

Jurassic News promuove la libera circolazione delle idee

Gennaio 2010

In allegato il compilatore PL/Bit per Apple II

In Copertina

Il menù di start-up di un sistema Indigo della Computer Graphics, una workstation UNIX di alto livello che ha puntato sulla grafica la sua offerta di sistemi ad alte prestazioni.

La clip con floppy Apple indica che a questo fascicolo è allegato un supporto magnetico. Per scaricarlo visitare il sito della rivista.

Editoriale

Signori: si cambia!

Era il gennaio 2006 quando usciva timidamente il numero uno della nostra rivista. Da allora una corsa ininterrotta di ventisei numeri che hanno impegnato non poco, come potete ben immaginare, il nostro gruppo redazionale e qualche amico che ci ha aiutato nel corso della nostra storia.

Come regola interna ci siamo dati la limitazione del mandato di coordinatore editoriale, una figura manageriale che è essenziale per riuscire a fornire in maniera regolare la rivista. Al coordinatore spetta "tirare le fila del discorso" pianificando prima e sollecitando poi la consegna dei vari contributi, arrivando alla "minaccia" per i ritardatari. Un compito che richiede molto tempo libero e che sarebbe giusto fosse diviso fra più persone.

Dopo l'esperienza di start-up fatta da Tullio, anche Salvatore sarebbe giunto al termine del suo impegno, ma sostituirlo non è certo facile e d'altra parte, pur avendo l'amico Salvatore proposto un coordinamento ad interim fino alla decisione di chi se ne farà carico, riteniamo non sia giusto approffittare troppo.

Qualcuno poi non se la sente, soprattutto per ragioni di lavoro, così tenteremo quest'anno un esperimento di coordinamento a rotazione. Comincerò io (Tullio) e poi seguiranno Sonicher, ancora Salvatore, Lorenzo e Bes.

Non è che questa cosa "interna" avrà una ripercussione diretta sui contenuti della rivista, ma forse ognuno porterà un qualche suo apporto per arricchire ulteriormente l'offerta di contenuti e di fruizione del periodico.

A questo proposito rimane sempre valido l'invito di farci pervenire le vostre osservazioni o i vostri "desiderata"; terremo conto di tutti i suggerimenti, come del resto abbiamo sempre fatto, nel limite ovviamente delle nostre possibilità.

Buona lettura a tutti.

[Tn]

Jurassic News

è una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte (e attentamente vagliate) da Internet.

Normalmente il materiale originale, anche se "jurassico" in termini informatici, non è privo di restrizioni di utilizzo, pertanto non sempre è possibile riportare per intero articoli, foto, schemi, listati, etc…, che non siano esplicitamente liberi da diritti.

La redazione e gli autori degli articoli non si assumono nessuna responsabilità in merito alla correttezza delle informazioni riportate o nei confronti di eventuali danni derivanti dall'applicazione di quanto appreso sulla rivista.

Retrocomputing

Quello che ci siamo persi...

Apple, Commodore, Sinclair, Atari,... va bene, ma gli altri?

Quanti calcolatori sono stati costruiti e dei quali abbiamo sì e no una flebile traccia?



Mi è capitato di recente di venire ad una discussione abbastanza accesa con un amico retro computerista, in merito alle macchine che sono state progettate e costruite e delle quali non abbiamo più traccia.

Il mio amico sosteneva che queste non potessero essere poi moltissime e che semplici progetti hobbistici o poco più non conservano grande interesse per il retro computing. La mia posizione era diversa e, se vogliamo dirla tutta, completamente antitetica alla sua: non solo ne sono stati costruiti molti di queste calcolatori, ma il fatto che non non diano più segno di se non li esclude dall'essere oggetti interessantissimi per il nostro hobby. Anzi, si potrebbe dire che la rarità ne costituisce il pregio.

:-"Stai parlando dei vari Compucolor, TRS-80, Amico-2000, etc...?" - mi chiede. A parte il fatto che il TRS-80, sebbene poco diffuso da noi è tutt'altro che una macchina sconosciuta, gli spiegai che mi riferivo piuttosto a quei calcolatori che sono stati costruiti da piccole aziende, magari comprando i componenti qua e la e realizzando software verticali come i gestionali e

Nelle immagini che illustrano l'articolo, alcune riproduzioni delle pagimne pubblicitarie della rivista Bit numero 20 del 1981.

poco altro.

Simili software-house (vogliamo chiamarle così?) spesso vendevano sistemi chiavi in mano con pochissimo software: giusto magazzino e fatturazione, cose così, messe insieme o adattate alle specifiche esigenze del cliente.

Passando dalle parole ai fatti abbiamo sfogliato assieme una vecchia rivista di informatica: Bit, esattamente il numero 20 che è del settembre '81.

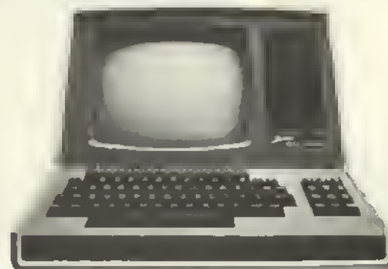
Abbiamo così scoperto una specie di cimitero dei dinosauri: soprattutto la pubblicità è una fonte rivelatrice dell'esistenza (fossile?) di decine di calcolatori ormai persi nell'oblio.

Sarebbe bello poter incontrare qualcuno che ce l'ha ancora in casa un "Ulisse-80", uno "Zenit data System", un "SuperBrain", un "Vector System 3005", un "Monroe EC4000", un "Saga Fox", oppure addirittura un "ABC-26" della AI Electronics Corporation Japan!

Qualcuna di queste macchine si può trovare anche su Internet; è il caso del SuperBrain, evidentemente bastantemente diffuso oltreoceano da aver generato un filone tutto suo nelle collezioni di retro calcolatori. Di altre, soprattutto se costruite o solo assemblate in Italia, inutile cercarne traccia: vuoto assoluto!

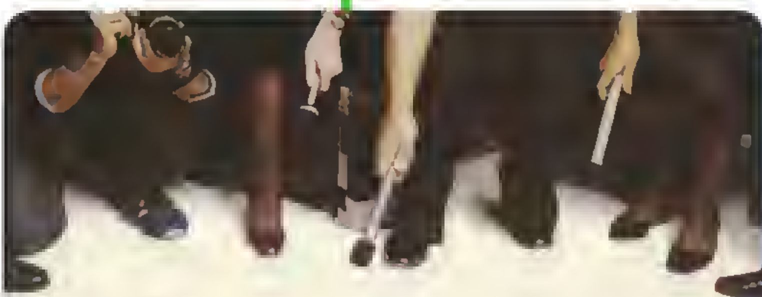
Quasi tutti questi progetti si basano sulla CPU Z80. Questo è espressamente dichiarato nelle specifiche tecniche oppure lo si deduce dalla presenza del CP/M o anche semplicemente dalla lista dei linguaggi disponibili. La comparsa di un FORTRAN-80 o di un ALGOL-80 nella lista dei sistemi di sviluppo acquistabili con la piattaforma, è chiarificatrice della tipologia di macchine offerte.

Alcuni sistemi, come lo Zenit Data System e il Superbrain, dichiara la presenza di due microporocessori Z80, chissà per che farne... immagino che uno sia dedicato all'I/O visto che non sembra il caso di ipotizzare configurazioni multiprocessore nel









1981!

Per altri progetti è più difficile dedurne la configurazione tecnologica. Ad esempio il Monroe EC 4000 non svela assolutamente nulla di particolare se non la dotazione di memoria RAM disponibile.

I nomi di fantasia, dallo stile palesemente scientifico (Compubionics, ad esempio), sempre completi di sigle numeriche, fanno intendere di essere difronte ad una macchina proiettata nel futuro.

La multiutenza è una frontiera: -"Il computer CTL 980 si fa in due..." che vuol dire semplicemente che ci potete attaccare due terminali. Sistema operativo e altri particolari non è dato saperlo (im maginiamo sia basato su MP/M, l'unico sistema operativo in time-sharing disponibile per i micro.

Beh, al termine di questa nostra carellata ci siamo trovati d'accordo: -"Ce ne siamo persi parecchi, peccato!".

[Tn]

Accesso casuale

Il FOX della SAGA



La SAGA ha appena lanciato il FOX un nuovo "small business computer" made in Italy.

Il sistema è piacevole esteticamente con scocca comprendente tastiera, video e due floppy da 5 pollici e un quarto.

L'architettura, monoscheda, è basata su due microprocessori Z80 con ciclo da 4 MHz di cui uno dedicato alla gestione dei dischetti e l'altro come unità centrale: ne deriva una notevole velocità complessiva. La memoria è standard da 64 Kbyte più 2 Kbyte EPROM di firmware.

Presenta in uscita un porta seriale RS-232 ed un porto parallelo utilizzato in linea di massima per la stampante.

La tastiera è particolarmente curata. Si presenta con la disposizione standard QWERTY, tastierino numerico separato con tasto di triplo zero. Protetta contro gli ingressi multipli, dotata di click controllabile a software, presenta alcuni tasti illuminabili sia per il controllo di POWER ON, del RUN delle periferiche, dello stato di Maiuscolo, sia per segnalazioni e programma. L'accoppiamento fra tasti e caratteri è interamente controllato a firmware. Ciò è particolarmente utile per quattordici tasti funzione che completano la tastiera con una riga aggiuntiva in alto destinati sia ad un uso di Word Processing sia a personalizzazioni varie.

Il video da 12 pollici, di colore verde a con trattamento antiriflesso particolarmente efficiente, presenta 24 righe da 80 caratteri maiuscoli e minuscoli ed un set di caratteri speciali modificabili a firmware.

La memoria di massa prevede due floppy ciascuno da 400 Kbyte (doppia faccia e doppia densità) oppure un floppy ed un disco fisso di tipo Winchester da 5,8 Mbyte.

Il software è particolarmente curato ed abbondante.

È disponibile il complesso delle procedure gestionali fornite "chiavi in mano" dalla SAGA in linguaggio (PL80). Ciò sia per la versione a floppy sia per quella a winchester. È annunciata un'estensione del BASIC che consentirà all'utente l'accesso agli archivi del software SAGA.

Il software di base elenca il sistema operativo CP/M, il complesso dei linguaggi disponibili sullo Z80 (Basic, Fortran, Cobol...) e vari emulatori per telecomunicazioni.

La SAGA sta curando la messa a punto di un pacchetto di Word Processing che prevede l'aggancio a routine di calcolo e ad archivi; ciò allo scopo di collegare con il WP procedure di tipo EDP.

È annunciato un pacchetto, (SCIS - Saga Colloquial Information System) tendente a proseguire una tradizionale sporta della SAGA verso l'"office automation". In esso sono particolarmente curate le funzioni di introduzione dati, di ricerca negli archivi e di preparazione di documenti a mezzo di un linguaggio di notevole semplicità.

Il sistema è costruito con materiali e periferiche quasi totalmente italiani e tutta italiana è la progettazione; le note specifiche che derivano direttamente dalla esperienza SAGA, sia nell'hardware, è comunque evidente ad occhio attento allo sviluppo della tecnologia internazionale.

Il prezzo di vendita è di 7.35 milioni nella configurazione con 64 Kbyte e due floppy da 400 Kbyte.

La SAGA, con un volume di affari di circa 10 miliardi, 70 dipendenti diretti, due filiali, quattro consociate ed oltre 35 concessionari ha svolto fino ad ora prevalentemente una politica rivolta all'EDP gestionale "chiavi in mano". Sta differenziando le sue attività da circa un anno con il word processing (distribuisce l'americano NBI) e con i sistemi grafici e colori ad alto livello (distribuisce il "tastini scan" della Ramtek).

Con il lancio del FOX intende rivolgersi anche ai system builders, alle software houses ed al mercato dell'informatica distribuita.

Per ulteriori informazioni:
SAGA
Via V. Bellini, 24
00198 Roma
Tel. 06/867741

136

Le prove di Jurassic News

Silicon Graphics - Indigo

Quando l'informatica personale ha cominciato davvero a incidere dove prima non era immaginabile arrivare, allora nacquero delle aziende che puntavano sulla qualità dei loro prodotti. Una di queste fu Silicon Graphics.



Già dal colore si capisce che questo non è un normale PC vestito da workstation... E' l'Indigo di Silicon Graphics, praticamente una leggenda!

Contesto storico

Prendete un profondo respiro, chiudete gli occhi e fate un rewind mentale fino al 1991. Ci siete?

Facciamo mente locale. Il Presidente del Consiglio è Andreotti. Lo scudetto lo vince la Sampdoria, il mondiale di Formula 1 Ayrton Senna. Edwige Fenech presenta il festival di Sanremo. In Kuwait scoppia la prima guerra del golfo. I Metallica incidono Nothing Else Matters. Muore Freddy Mercury.

Il mercato degli home computer è saldamente in mano alle varie versioni dell'Amiga. Nel settore dei PC la Intel ha già presentato il 486, ma sul mercato impazzano i 386SX a 20-25 MHz che permettono alle ditte di smaltire tutta la vecchia componentistica AT a 16 bit. Ci sono ancora in uso moltissimi 8086 e 80286, e gli sviluppatori aspettano il momento in cui ci sarà un 386 su ogni scrivania. Gli hard disk hanno una capienza tipica intorno ai 40 MegaBytes. La Apple propone il Macintosh LC, con un Motorola 68020 a 16 MHz e grafica a colori da 512x384 pixel. La Microsoft presenta la versione 5.0 di MSDOS, limitata come sempre a 640 Kbytes, mentre Windows è ancora alla versione 3.0, che è poco più di uno shell grafico e ha un con-

corrente temibile in OS/2; tra l'altro non supporta in alcun modo la rete locale e non lo farà fino al 1993. Internet non è ancora il www (che viene presentato proprio quest'anno) e viene usato solo dagli addetti ai lavori. Esistono invece i BBS, a cui si accede via telefono e modem; naturalmente non i modem di ieri a 56 Kbaud ma quelli di allora, a 300 o 1200 baud. Nella grafica si sono ormai imposte le schede VGA (640x480 con 16 colori, nessuna accelerazione 3D), che vengono prodotte in modelli incompatibili da una infinità di piccole ditte (Orchid, Paradise, Cirrus ecc.ecc.). I drives CD-ROM non esistono e le schede audio sono ancora rare.

Naturalmente per usi professionali esistono, come sempre, macchine di categoria superiore. Dopo la prova generale con The Abyss (1989), la computer graphics entra infatti quest'anno nella storia del cinema con Terminator 2 mentre è in lavorazione la pietra miliare di Jurassic Park, che uscirà due anni dopo. Dietro gli effetti speciali cinematografici troviamo in pratica una sola ditta: la Silicon Graphics. In particolare il suo modello più legato alla produzione di Jurassic Park verrà prodotto in seguito, con una certa ironia, anche nella versione limitata Jurassic Classic.

La Silicon Graphics Inc.

La Silicon Graphics è da sempre specializzata in sistemi professionali high-end. Nel 1988 ha fatto con la sua Personal Iris un primo tentativo di introdursi in una fascia di mercato più bassa, riuscito solo parzialmente: la Personal Iris è impressionante ma costa ancora molto, almeno 25.000 U$. Ora, nel 1991, presenta una nuova macchina per lo stesso target. Il costo è ancora lontano dal mondo PC, partendo da un minimo di 10.000 U$ dell'epoca, ma non assurdo per un uso professionale. E un confronto coi PC coevi è impossibile: già la prima versione ha un processore RISC MIPS R3000 a 33 MHz (ma sarà subito affiancato dal nuovo R4000 a 100 MHz) con CPU+FPU, monta fino a 16 Mbytes di RAM e fino a tre Hard Disk SCSI veloci per una capacità di diversi Gbytes. La risoluzione grafica è fissa a 1280x1024 pixel, con 16 milioni di colori e accelerazione hardware. Il suono, stereo, è



*(Figura 1)
Negli anni '80 e '90 BYTE era la più autorevole rivista USA per il personal computing. In copertina nel numero di Settembre la Indigo in tutto il suo splendore. Da notare la preview di Windows 3.1!*

affidato ad un Digital Signal Processor Texas Instruments 56000. Un'interfaccia thick-ethernet AUI è di serie. Il sistema operativo è una variante di Unix stabile ed efficiente, con una bella interfaccia grafica WIMP (windows-icons-mouse-pointer).

E' la Silicon Graphics Indigo, che conquista subito la copertina di BYTE del Settembre 1991.

Primo approccio

La Silicon Graphics non ha mai, in tutta la sua storia, prodotto una macchina beige. La Indigo è appunto blu scuro (Indaco!), compatta e squadrata, percorsa da sottili feritoie di raffreddamento. Larga 25cm, alta 38 e profonda 30, la Indigo è a parere di molti la macchina più bella mai prodotta da Silicon Graphics – che non è poco, considerata la qualità dell'industrial design di questa casa, paragonabile solo a quello di Apple. Anche la qualità costruttiva non ha niente in comune coi PC: la macchina, come quasi tutte le Silicon che seguiranno, è interamente contenuta in un telaio d'acciaio che sembra in grado di resistere ad un rientro in atmosfera, ricoperto da un guscio esterno in plastica che ha solo funzioni estetiche e che la Casa chiama con sufficienza skin, pelle. Con un solo hard disk montato pesa già 17 Kg.

Uno sportello molto discreto permette l'accesso ai dispositivi rimovibili (Figura 2). Nel caso della macchina in prova l'unità raggiungibile aprendo lo sportello frontale è un DAT montato sullo slot superiore.



(Figura 2)
Uno sportello molto discreto permette l'accesso ai dispositivi rimovibili.

Il badge giallo in basso a sinistra indica che la macchina è una Indigo R4000 Elan. In teoria tutte le R4000 avevano una badge colorata di identificazione, di forme e colori diversi a seconda delle schede grafiche installate. In pratica quasi tutte le Indigo sono state modificate in qualche modo, non sempre coerente con l'aspetto esterno.

A parte un backplane di collegamento, che contiene anche le interfacce parallela e SCSI, tutta l'elettronica è concentrata in due grandi schede a innesto rapido. Una contiene la CPU, la memoria, la rete, tutte le interfacce tranne l'uscita video e due slot di espansione proprietari; la seconda è la

scheda grafica con i suoi connettori. Il pannello frontale si smonta con due pulsanti e all'interno non c'è un solo filo volante: tutto, alimentatore incluso, è montato su slitte ad innesto diretto sul backplane, e tutto si smonta e si rimonta senza attrezzi. Avendo i ricambi adatti, una Indigo guasta può essere rimessa on-line in due minuti e a mani nude.

Hardware

In realtà la configurazione che abbiamo descritto è solo una delle infinite possibili. La Indigo è da considerarsi più un sistema che un modello, e sotto una identica apparenza esterna possiamo trovare strutture completamente diverse.

La scheda CPU+FPU, che inizialmente ospita un processore R3000 a 33 MHz, viene subito ridisegnata in modo da accogliere il nuovo processore R4000 a 100 MHz (e poi il successivo R4400 a 150 MHz) e molta più RAM. Può montare fino a tre hard disk, ma due di questi possono essere sostituiti da un floppy da 3,5" e/o da uno speciale floppy magneto-ottico (il floptical) da 21 MBytes e/o da una unità a nastro DAT da 2 GBytes. La scheda grafica può anche mancare del tutto se la macchina



*(Figura 3)
Rimuovendo il frontale si nota a sinistra lo stretto sportello verticale che accoglie la scheda CPU e la scheda grafica.*



*(Figura 4)
Lo sportello schede si apre verso il basso rivelando il fianco delle schede inserite negli slot.*



(Figura 5)

Il pannello posteriore con le uscite "built-in" e quelle messe a disposizione dalle schede inserite.

viene usata come server, o può essere una scheda entry-level piuttosto spartana, ma più spesso è la scheda modulare GR. Questa è basata su diversi PCB e sembra fatta coi Lego: può essere equipaggiata (e/o espansa successivamente) con uno o con tre piani di RAM, per avere rispettivamente 256 colori o 16 milioni di colori; può avere o non avere uno Z-buffer per assistere il rendering 3D; può montare uno, due oppure quattro Geometry Engines che eseguono in hardware le trasformazioni geometriche, più un acceleratore raster (il Raster Engine) che è sempre presente; è predisposta per il video stereo 3D in hardware. La versione "full optional" di questa scheda, con 24 bpp, Z-buffer, Raster Engine e quattro Geometry Engines si chiama Elan, e fa una certa figura ancora oggi. Una conseguenza di questa componibilità è che oggi è difficile trovare due Indigo uguali, perfino provenienti dallo stesso ufficio.

Nella figura 2 è possibile notare a destra, dall'alto, l'alimentatore (su cui sono fissati anche l'altoparlante e il grosso pulsante di reset) e i tre slot per dischi e nastri, tutti montati su slitta.

Nella figura 3 la macchina aperta mostra le schede inserite negli slot. Nella scheda video, quella più a destra, si notano alcune delle "daughterboards" che costituiscono la configurazione Elan. Uno dei dischi fissi è parzialmente estratto.



(Figura 8)

Fratelli? Gemelli!

La Indigo infatti è stata venduta – senza alcuna modifica se non la marca e il colore - anche dalla rete commerciale di altri costruttori: in questo caso Control Data Corporation e Siemens-Nixdorf.

Il pannello posteriore (Figura 6) non è particolarmente ordinato. Nella scheda video (la serie di connettori a sinistra) troviamo, dell'alto, il connettore 13w3, un BNC per il genlock e un mini-DIN per gli occhiali stereo CrystalEyes.

A fianco, cinque connettori mini-jack per microfono, linea in, linea out, cuffie e audio digitale (!); due porte seriali RS232 con connettori mini-DIN compatibili Macintosh, un terzo mini-DIN per la tastiera ed il 15-pin della rete thick ethernet.

Più in basso il DB-25 della porta parallela e al centro il grosso SCSI 50-pin. I due pannelli metallici più piccoli a destra, coprono gli slot di espansione dedicati.

Per fare a meno del video con uscita 13w3 e collegarci un VGA, è necessario disporre di un adattatore come quello mostrato nella figura 7 assieme all'adattatore AUI-RJ45 per utilizzare l'attuale standard di collegamento per la rete ethernet.



(Figura 6)

Un dettaglio della scheda CPU di una Indigo R3000. I due chip dorati NEC sono la CPU R3000A-33 e la FPU R3010A-33, che non hanno bisogno di dissipatori.

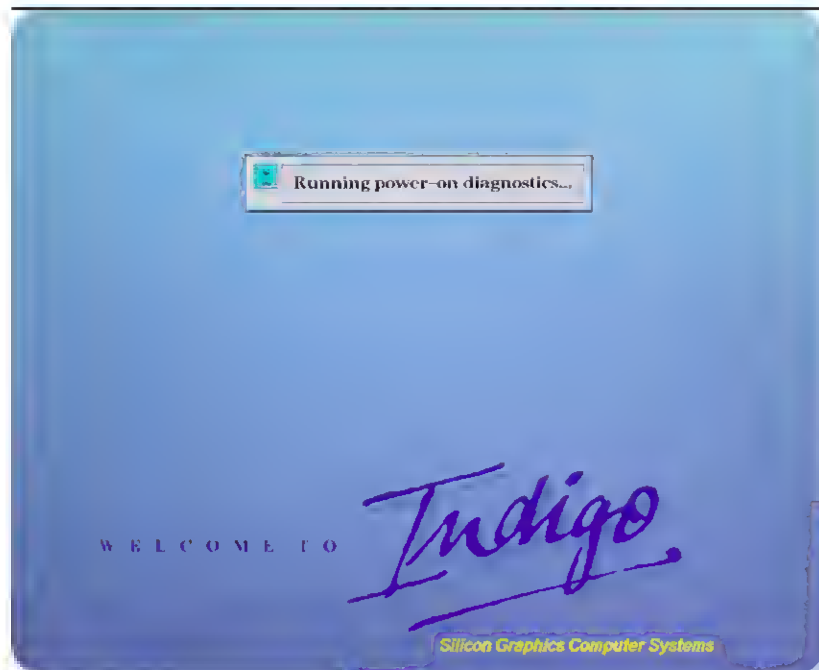
Software

Il sistema operativo si chiama Irix, è la versione Silicon Graphics di Unix e segue piuttosto da vicino la varietà AT&T piuttosto che la BSD favorita da altri costruttori. Naturalmente ne esistono molte releases successive; in origine le Indigo montavano Irix 4.0, ma praticamente tutte sono state aggiornate a qualche versione successiva. Le Indigo con processore R3000 possono accettare soltanto fino alla versione 5.3; sulle R4000 è più comune trovare la 6.2, ma una R4000 Elan può reggere brillantemente fino alla release 6.5.22 del 2003 (adesso pensate a come girerebbe Windows 2003 su un PC del 1991, e ridete). L'ultima versione di Irix, gestibile solo da macchine successive, è del 2006.



(Figura 7)

Adattatore video e retesono due accessori ormai indispensabili per chi ha un Indigo e vuole usarlo con le attuali tecnologie video e di rete.



(Figura 9)

La schermata iniziale di benvenuto.



(Figura 10)

Il menu di servizio cui si accede se si interrompe il caricamento del sistema operativo.

Uso

All'accensione la macchina emette un caratteristico motivo musicale e lancia il self-test (figura 9). Le Silicon Graphics non hanno un BIOS; all'accensione eseguono un programma in ROM che provvede al caricamento di Irix dall'hard disk e permette, volendo, di interrompere l'avvio per operazioni di manutenzione quali aggiornare il sistema operativo, eseguire un ripristino del sistema, effettuare una verifica approfondita dell'hardware o caricare un interprete di comandi provvisorio. Il caricamento di Irix impiega qualche minuto, ma occorre ricordare che questa non era pensata per essere una operazione frequente e che anzi la casa consigliava esplicitamente di lasciare sempre accesa la macchina, in modo che durante la notte potesse eseguire automaticamente determinate operazioni di housekeeping. Infine viene caricato un menu grafico che elenca sotto forma di icone gli utenti autorizzati al login (figura 10).

Fatto il login appare il desktop, che presenta in alto a sinistra il menu (chiamato Toolbox) delle operazioni più comuni e una serie di icone allineate lungo il margine destro. Nel Toolbox sono presenti, tra le altre cose, l'Icon Catalog che raccoglie in cartelle separate le applicazioni installate (di serie è inclusa una ricca serie di utilities e parecchi demo, tra cui il gioco Doom) e un System Manager grafico che dà una interfaccia visuale a tutte le funzioni di amministrazione, incluso il controllo completo dei processi, del filesystem e la creazione e la gestione di volumi logici estesi su diversi dischi fisici.

Ovviamente colori, sfondi, fonts e persino le dimensioni delle icone sono personalizzabili. Bisogna ammettere che per lo sfondo desktop sono disponibili poche possibilità, quasi tutte di rara bruttezza, e persino nelle versioni più recenti non è previsto in alcun modo che l'utente aggiunga sfondi extra. In realtà

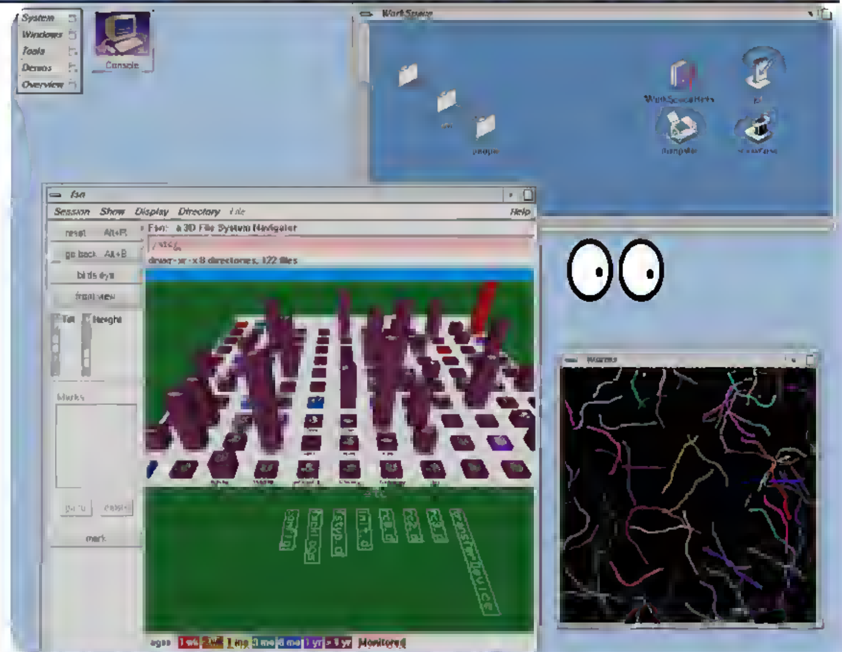
questo è possibile, con un piccolo hack che per i motivi appena esposti si impara molto presto.

Irix è, naturalmente, molto stabile e concepito per girare ininterrottamente 24/7. Ha tutti i soliti vantaggi di Unix con una serie di utilities esclusive che lo rendono molto comodo da amministrare: per fare due soli esempi, il comando chkconfig che permette di controllare, avviare e fermare individualmente i daemons con una facilità disarmante, e un sistema di installazione del software a pacchetti pressochè identico ai package manager attuali di Linux.

La documentazione è eccellente e tutti i (molti) manuali disponibili, che su carta riempirebbero uno scaffale, sono preinstallati in formato elettronico. La GUI è garantita da X-Windows e dal window manager proprietario 4Dwm.

Irix richiede un mouse a tre tasti; non riconosce la rotellina, ma in cambio può montare diversi dispositivi di input tra cui manopole, arrays di pulsanti e il particolare Spaceball a sei assi.

In qualche punto l'hardware della Indigo dimostra tutta la sua età. È forse l'ultimo computer ad avere un "vero" interruttore di accensione: tutte le Silicon successive avranno infatti non solo lo spegnimento via software a cui siamo oggi abituati, ma anche la capacità di riaccendersi autonomamente ad un orario programmato.



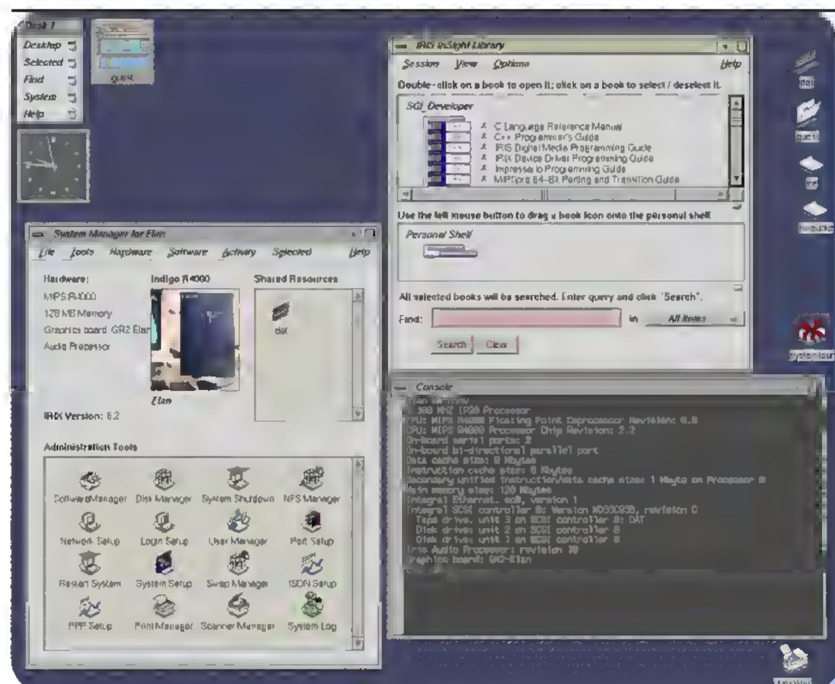
La scheda video GR non solo usa uno speciale connettore detto 13w3 che richiede un adattatore per essere collegato ad un monitor moderno, ma trasmette i segnali di sincro sovrapposti al segnale del verde (SOG, sync-on-green).

Il connettore 13w3, che prevede tre contatti RGB e dieci contatti ausiliari, era abbastanza diffuso anche tra altri costruttori (SUN, HP, IBM ecc.) ma non è mai stato completamente standardizzato e la funzione dei contatti ausiliari varia non solo da marca a marca ma persino da modello a modello, cosicché un adattatore non-specifico potrebbe anche funzionare come anche no.

Un vero pericolo poi è il connettore della tastiera (alla quale si collega in cascata il mouse), che fisicamente è un mini-DIN PS/2 ma ha voltaggi e connessioni incompatibili. In questo modo è possibile connettere per errore alla Indigo una tastiera PC, con conseguenze tragiche.

(Figura 11)

Il desktop, ancora piuttosto povero, di Irix 4.0. Il programma grafico 3D in primo piano (che non è di serie) è l'applicazione originale che appare nelle scene finali di Jurassic Park (ricordate la scena? "Questo è un sistema Unix... io lo conosco!").



(Figura 12)

Il desktop di Irix 6.2. In evidenza il system manager 6.2 e, a destra, il sistema di documentazione InSight con i numerosi manuali on-line.

Il resto è storia...

Cosa ne è oggi della casa produttrice? Sotto molti aspetti la Silicon Graphics ha semplicemente fatto la storia della Computer Graphics. Negli anni successivi ha cambiato nome in SGI e complessivamente ha prodotto molti altri modelli, alcuni dei quali, come la Indy e la O2, sono stati un successo planetario (curiosamente sono anche le macchine meno performanti). All'altro estremo della gamma vanno ricordati modelli incredibili come il supercomputer grafico Onyx2, in grado di ospitare fino a 64 CPU e tre canali video Infinite Reality multiplexati in grado di comporre su tre megaschermi affiancati un'unica immagine panoramica in stereo.

I processori MIPS ed i loro compilatori hanno mantenuto prestazioni elevatissime ma alla fine hanno perso la battaglia dei costi, e le ultime macchine MIPS-based sono uscite di produzione nel 2006 insieme all'ultima release di Irix. Come HP e Apple anche SGI è passata alla piattaforma Intel, e a Linux come sistema operativo.

Dopo un difficile periodo di amministrazione controllata la SGI si è riconfigurata come fornitrice di sistemi di memoria e server di alta classe e alta affidabilità. La grafica ha rivestito un ruolo sempre più marginale nella sua attività finchè, nel 2009, l'intera sezione grafica è stata chiusa. Sic transit gloria mundi.

Sul fronte software la SGI ci ha lasciato, tra l'altro, lo standard OpenGL, e dobbiamo alla sua opera pionieristica le schede grafiche dei PC attuali: Nvidia è stata fondata da ex-dipendenti SGI.

In retrospettiva la Indigo è stata una macchina capace di prestazioni notevoli ma anche, per certi versi, una macchina di transizione che dopo solo due anni si è sdoppiata nei modelli Indigo2 e Indy. Inoltre (e purtroppo) oggi il suo case grande ed elegante è anche appetito dai modder, che lo svuotano per inserire al suo interno un PC moderno. Come tutti i retrocomputer non ha un valore di mercato definito: chi scrive se ne è vista offrire una gratis e un'altra (incompleta!) a oltre 200 euro su Ebay. Ma forse la cosa più sorprendente è che alcuni rivenditori/appassionati mantengono un vero e proprio magazzino di diversi modelli Silicon Graphics e relativi

componenti, il che rende ancora possibile, dopo quasi vent'anni, ordinare una Indigo nella configurazione preferita come si farebbe con un normale PC. La SGI ha infatti un seguito di entusiasti piccolo ma accanito, persino incredibile per macchine che non sono quasi mai state vendute direttamente al pubblico.

Nel giro dei retro-computeristi fedeli a questa marca viaggia ancora molto freeware e shareware aggiornato e, volendo, una Indigo sarebbe ancora perfettamente usabile per svariate applicazioni (o per navigare, installando una versione recente di Firefox).

C'è tuttavia un uso in particolare che ci sentiremmo di suggerire: se ci si riesce a procurare e installare un programma per la riproduzione di files mp3, il vecchio DSP Texas 56000 della sezione audio fa ancora paura e non occorre essere un audiofilo raffinato per sentire la differenza di resa sonora con un PC Windows. Provare per credere.

[Mr]



(Figura 13)

Ancora il desktop 6.2 (ma il più recente 6.5 è pressoché identico) con il raccoglitore Icon Catalog aperto sulla tab Demos. In alto a destra il "demo di demo" buttonfly, che era un vero e proprio biglietto da visita per Silicon Graphics. Ricordo che ne esisteva anche un clone per Amiga – per la serie "vorrei ma non posso".

Il racconto

Rubrica curata da [Bs]

Storie di vita dove i computer (soprattutto retro computer) c'entrano in qualche modo.

Automatik(3) - La bisca

Una sera il titolare della ditta dove ero ospite e stagista mi venne a trovare nell'alberghetto dove alloggiavo, almeno questo a spese del mio futuro datore di lavoro, e mi chiese se potevo l'indomani dare una mano a lui e ai ragazzi per andare poco lontano a ritirare della "roba", così disse. Naturalmente accettai senza che mi passasse nemmeno per l'anticamera del cervello che questa attività poco aveva a che fare con un mio allenamento all'elettronica, ma tant'è la mattina si partì presto, saranno state le sette, con due furgoni alla volta di una imprecisata località sul Garda.

Arrivammo nella nebbia autunnale in un posto che mi sembrò dimenticato da Dio dove stava una specie di stamberga, almeno così mi parve, con una scritta che recitava "Albergo delle Rose", malmessa essa stessa, tanto che ricordo perfettamente che il nome lo ricavai non già dalla presenza delle lettere dell'insegna, quanto dal colore diverso che aveva la tinteggiatura in corrispondenza di quelle mancanti che erano la maggior parte.

Qui in uno scantinato raggiungibile con una doppia rampa di scale strette di grezzo cemento, stavano depositati tre biliardi, uno smontato e due interi. L'ambiente la diceva lunga sulla destinazione d'uso di un simile locale, evidentemente abbandonato in tutta fretta per qualche motivo: divani di vistosa plastica rossa e gialla (dico davvero, gialla!), tavolini coperti di panno verde, adibiti evidentemente al gioco, sedie di una qualche velleità di comodità visto che erano imbottite di un velluto viola difficilmente accostabile a qualsiasi arredamento che non fosse adatto ad un bordello. Michele suggerì che doveva essere lo stesso posto oggetto di una irruzione della Polizia un paio di anni prima. Sembra si trattasse di un vero e proprio covo dove avveniva di tutto, comprese partite a biliardo evidentemente, non saprei dire con quale scopo o posta.

In quel posto il nostro titolare svelò lo scopo del viaggio: -"Portiamo via questi due biliardi e quello smontato che c'è lì", boforchiò, indicando con un cenno del capo un cumulo che a me sembrava di macerie ma che evidentemente ad un esame più attento evidenziavano immancabilmente l'origine. "Portiamo" aveva detto il Nostro Duce, ma non seguì alla lettera la sua stessa dichiarazione, nel senso che NOI portammo su la roba, lui sparì con la persona dall'aria equivoca che

aveva aperto la porta del tugurio.

Non so quanto siate pratici di biliardi ma certo avrete avuto la sensazione toccandone uno che siano molto massicci. E' così! Visto che il piano deve essere liscio al massimo e assolutamente stabile, ne segue che è realizzato mettendo assieme a seconda della dimensione del piano stesso, da quattro a otto lastre di pesantissima ardesia levigata della dimensione di circa un metro quadrato cadauna e spessore dai quattro agli otto centimetri. In tre non riuscimmo ad alzare la prima che stava appoggiata al muro e subentrò lo sconforto. Che fare? Non eseguire sarebbe valso ad un ammutinamento in piena regola con conseguenze spiacevoli per tutti e tre (Salvatore mica era venuto, lui era troppo furbo per farlo!).

Disponevamo di un carrello a due ruote, più adatto al trasporto delle cassette di frutta che di un simile peso. Quando riuscimmo a trascinare la prima piastra sul carrello le ruote dello stesso si abbassarono di una buona metà, a riprova che non c'era nulla da scherzare.

In ogni caso eravamo tre baldi giovani: Michele, Andrea e io. Certo la forza non ci mancava, per cui con rara abnegazione che nessuno al giorno d'oggi crederebbe possibile, trascinammo il primo pesante fardello su per le due rampe di scale e da li dentro il furgone. Rimanevano altre cinque piastre dello stesso peso, poi le gambe e le fiancate e poi due altri biliardi interi, e per quel primo viaggio avevamo impiegato mezz'ora buona!

Il secondo viaggio andò peggio, nel senso che la piastra si scheggiò in un angolo, il terzo fu parimenti faticoso e a quel punto noi eravamo assolutamente esausti e certi di non essere in grado di portare a compimento l'arduo compito. Ci sedemmo quindi sulle scale di cemento per prendere fiato e naturalmente, dato che il diavolo ci mette sempre lo zampino, capitò subito il titolare assieme ad una tizia dall'aria volgare, anche se giovane e certo non brutta come fisico, che fumò tutto il tempo una sigaretta dopo l'altra mentre si accalappiava sul prezzo che il nostro capitano avrebbe dovuto sborsare per liberarla di quel pesante ingombro.

Noi naturalmente facevamo finta di proseguire la lotta con l'improbo mostro attaccando la piastra numero quattro del nostro calvario. Tiziano prima finse di darci una mano e poi si rivolse alla sventurata ciminiera umana per continuare le sue trattative al ribasso sul prezzo del materiale. Arrivati a metà della scala il Nostro ci intimò di riportare tutto indietro, evidentemente per le conseguenze del mancato accordo sul prezzo. Il tira e molla andò avanti per un pezzo con noi che facevamo dietro front su quelle scale ora salendo ora discendendo in relazione all'umore della trattativa. Alla fine decidemmo di rimanere sul pianerottolo e di non muoverci più, cascasse il mondo!

Il mondo non cascò naturalmente e dopo un'ora ci rivide arrancare

con il nostro fardello su quella scala che divenne per me simbolo della più assoluta e inutile sofferenza umana.

Il titolare non partecipò all'impresa, se non chiamando il colpo al momento di caricare, per criticare come avevamo sistemato le piastre sul furgone e imprecare per la nostra lentezza nel compiere le operazioni che lui, a suo dire, aveva fatto centinaia di volte. Si accorse della piastra sbeccata e dato che le bestemmie pronunciate all'indirizzo di tutti i santi del Paradiso, noi compresi, non ebbero efficacia tale dal riparare il danno, volle di nuovo ridiscutere il prezzo con la megera di prima adducendone il motivo alla scoperta della magagna.

Da parte nostra eravamo arrivati a quello che credevo potesse essere il limite della sopportazione umana, senza sapere che l'uomo ha risorse tali e a lui insospette, da portarlo ben oltre questo limite.

La megera venne, si rese conto del danno e sentenziò che era stato fatto da noi medesimi e che la scheggiatura era recente. La sua competenza ci lasciò allibiti e ancora di più ne fummo sorpresi quando con astuzia rifece il percorso delle scale e individuò il punto esatto dove era avvenuto il fattaccio. Noi avevamo buttato il pezzo grosso della scheggiatura ma non certo raccolte tutte le schegge che si erano generate.

Io e Andrea stemmo zitti mentre Michele, anima candida, confessò confuso il nostro fallo e a noi non rimase che annuire.

La megera parlò tenendo la sigaretta in bocca e con soddisfazione che non si preoccupò minimamente di celare, ci redarguì come fossimo ragazzini che dicevano bugie. Pivelli lo eravamo di sicuro e quella inquietante, per noi novellini, figura femminile ci faceva un effetto particolare: ce ne stemmo zitti a testa bassa dopo aver confessato il peccato a quello che manco lontanamente si sarebbe potuto scambiare per un prete. Tacemmo e riprendemmo il lavoro, certi di non essere ormai più in grado di protestare alcunché.

La "signora", sempre con la sigaretta in bocca ci guardava arrancare per le strette scale e biascicava talvolta degli inviti a stare attenti e a non sbattere le sponde contro il muro. A me parse che ci prendesse in giro e lo dissi a Michele che ne convenne. Andrea disse che se era così lui se ne sarebbe andato, al diavolo le conseguenze (era giovane, lo perdonammo).

Poi la megera, così come era venuta, cioè fumando, salì le scale e sparì dalla vista. Lasciati i propositi di rivolta o incrocio delle braccia, trasportammo di slancio le ultime due piastre del primo biliardo e tutti gli altri pezzi che si rivelarono altrettanto pesanti e ingombranti su quelle strette scale.

Intanto era arrivato un tizio, evidentemente uomo di fiducia della padrona, che si mise con il nostro

titolare a smontare i due bigliardi ancora interi.

L'oggetto era un po' più piccolo di quello che avevamo caricato e di quello che rimaneva ancora integro nel locale, per cui ci illudemmo che sarebbe stato meno pesante: errore clamoroso! Il piano era di sole quattro piastre ma ENORMI! Dichiarammo senza pericolo di smentita che non ci sarebbero mai passati su per le scale e il nostro Duce non poté che constatarne la fondatezza di questa affermazione.

Ipotizzo di togliere le ringhiere in ferro per far transitare l'ingombrante fardello, ma convenne che avremmo dovuto segarle, dato che erano cementate alla base.

Il nuovo venuto si scrollò nelle spalle rivelando che c'era un'altra apertura sullo stesso livello di dove ci trovavamo, dove ci entrava anche il furgone intero, se lo volevamo.

Se lo volevamo?

Lo guardammo come dei naufraghi guardano la sottile linea all'orizzonte che qualcuno ha affermato essere la terra ferma. Se non fosse stato così torvo in viso e a dire la verità anche un poco sporco, l'avremmo abbracciato e baciato per la rivelazione che ci aveva testé fatta!

Detto fatto il tizio andò dietro una specie di palco e scostando dei tendaggi pesanti di un velluto rosso ricamato in oro, svelò un portone di ragguardevoli dimensioni con tanto di doppio catenaccio. Non era nemmeno chiuso a chiave: aprimmo e ci trovammo su un piazzale raggiungibile tramite una rampa che arrivando avevamo visto ma che pensavamo portasse a dei garage interrati.

La scoperta fu per noi una vera gioia e pur sapendo che sarebbe comunque stata una faticaccia il caricare tutto il materiale rimanente, almeno ci eravamo levati di mezzo le famigerate scale; la tortura della nostra mattinata.

Intanto si era fatta l'una e il nostro condottiero decise che aveva fame e che (forse) ce l'avevamo anche noi. Accettammo la pausa e l'invito, con discrezione ma con assoluta fermezza nel considerare la pausa necessaria in modo assoluto.

Mangiammo bene, devo dire ad onor del vero, in un ristorante sulla provinciale dove si fermavano sempre i camionisti: servizio celere, porzioni abbondanti e cibo gustoso. La nostra defaticante mattinata fu presto dimenticata e oggetto di lazzi a non finire sul fatto di come eravamo scemi a non aver capito che nessuno si sarebbe fatto quelle scale per portare i bigliardi nel seminterrato!

La megera venne con noi, non smise mai di fumare, lo fece anche mentre mangiava; era incredibile: masticava con la sigaretta accesa in un angolo della bocca!

Andrea, che era di un pezzo il più giovane fra noi tre, ne era affascinato e sbirciava le gambe della signora appena gli si presentava l'occasione. Sia io che Michele ci lanciavamo segni di intesa: per noi era chiaro come il sole che la tizia

era a dir poco equivoca e il suo parlare sboccato la qualificava indiscutibilmente. Il nostro titolare invece era a suo perfetto agio e oserei dire che la corteggiava in qualche momento, se la cosa non mi fosse sembrata di pessimo gusto.

Comunque il pranzo finì in bellezza con caffè e amaro doppio, il tutto signorilmente offerto dal nostro titolare. La megera si fece mettere in conto anche due pacchetti di sigarette dichiarando che: -"Poi facciamo i conti". Il nostro eroe pagò tutto e stette zitto, forse sperando di recuperare altrimenti le spese della sua magnificenza, lascio immaginare a voi come.

Non potrei giurare che la cosa gli riuscisse, ma quello che è certo è che sparirono entrambi "in ufficio" e riapparvero solo a cose fatte, cioè a furgoni caricati e noi impazienti di partire che si erano fatte quasi le cinque di sera.

Nell'attesa esplorammo quello strano locale scoprendo delle stanze attigue ben arredate con divani e con tavoli da gioco. Arrivammo poi in una stanza ampia dove c'erano un sacco di sedie imbottite con velluto verde, specchi alle pareti ma nessun tavolo o altro arredo che potesse rivelare a quale utilizzo essa era adibita.

Il grande lampadario che avrà avuto diametro di due metri, dal quale pendevano innumerevoli cristalli (non me ne intendo e non saprei se fossero preziosi o semplici riproduzioni di plastica), era appeso al soffito al centro della sale. Su tutto il pavimento una moquette rossa, anzi amaranto.

Michele mi indicò lo spazio vuoto sotto il lampadario e disse che era chiaro che li c'era stato un tavolo e mi mostrò i segni delle gambe dello stesso sulla moquette. Concordammo che era un tavolo da roulotte; forse non era vero, ma così voleva la nostra fantasia.

Quando il titolare della Game Star, Tiziano, si rifece vivo era ormai buio. Aveva un'aria soddisfatta, come quando in certe rare giornate ti va tutto per il verso giusto. Noi al contrario eravamo esausti, affamati e vogliosi di andarcene a fare la doccia.

Partimmo quindi io e il titolare su uno dei due furgoni, Andrea e Michele alla guida sull'altro. Arrivammo che erano quasi le sette e si doveva scaricare. Altra faticaccia, altre imprecazioni e altra scheggiatura di una delle piastre, questa volta in maniera seria, tanto da renderla praticamente inservibile, almeno questo disse il nostro capo quando si rifece vivo. Mica era stato li ad aiutarci a scaricare, ci mancherebbe! Si arrabbiò e molto, poi si rese conto che se avesse detto ancora una sola parola la piastra che stavamo scaricando in quel momento sarebbe finita sui suoi piedi. Fece un passo indietro, nemmeno mi avesse letto nel pensiero e se ne andò brontolando. Quella sera arrivai all'albergo che erano

quasi le dieci, ero stanco morto e per la cena era tardi. Mi buttai sotto la doccia e ci stetti finché l'acqua cominciò a venire fredda, poi mi addormentai di botto appena toccato il cuscino.

Il giorno dopo mi sentivo come se fossi saltato giù da un treno in corsa e rotolato trecento metri giù per la scarpata: letteralmente a pezzi. In laboratorio constatai che Michele e Andrea non stavano meglio di me e stavano concertando il modo di rifiutarsi in futuro di fare quelle sfacchinate. A me poco importava ormai: ero qualsi alla fine della terza settimana e fra qualche giorno si sarebbe compiuto il mio destino. basta bigliardi! Pensai ma mi sbagliavo dalla grossa. Infatti anche alla Automatik si noleggiavano e vendevano bigliardi, ma io non lo sapevo ancora...

Con il senno di poi mi viene spontaneo biasimarmi per aver accettato qualsiasi imposizione relativa al lavoro. Avevo troppa voglia di cambiare, di pavoneggiarmi con il titolo di "tecnico elettronico", di dimostrare con potevo fare qualche lavoro intellettuale invece che starmene a spazzare il magazzino del supermercato. Oggigiorno imparo dai giovani che non vale la pena affannarsi tanto: basta aspettare rifiutando qualsiasi offerta di lavoro che non ci sembri adeguata. Tanto, pagano i genitori...

Comunque imparai abbastanza cose in quel breve spazio di tre settimane passate alla Game Star e non solo cose tecniche, come si è visto.

La prima volta fuori casa per così tanto tempo, in un ambiente nuovo dove non conoscevo nessuno. Non è nulla, lo so, ma capii per la prima volta che si poteva fare, che ci si poteva anche muovere dal paese e guadagnare con un po' di sofferenza un bricciolo di conoscenza in più.

Anche le delusioni, le faticacce, il lavoro non dovuto e lo sfruttamento da parte di Tiziano che si era trovato per un giorno un operaio in più e gratis, ritengo facciano parte del mettersi in gioco e dell'accettare le esperienze che la vita ci manda, buone o cattive che siano.

[Lp]

Come eravamo...

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

Storia dell'interfaccia utente (2)

Tastiera e stampante

Se ci pensiamo bene, una cosa è rimasta più o meno intatta nel corso degli anni: la tastiera. La tastiera alfanumerica ha delle qualità evidentemente insostituibili e, salvo l'adeguamento tecnologico del caso, stiamo usando ancora un meccanismo di input che risale alla scoperta della macchina da scrivere.

Tastiera e stampante messe insieme nello stesso dispositivo (un tavolino di metallo), hanno costituito il primo terminale a caratteri preannunciando quella che sarebbe stata una evoluzione possibile: un colloquio riga per riga con la macchina mediato da un display elettronico. In un primo tempo l'output fornito dopo ogni riga emessa da tastiera, serviva unicamente come feedback per l'operatore che aveva così conferma dell'avvenuta digitazione corretta della riga di caratteri di input. Analogamente nell'output che praticamente abbisognava semplicemente della stampa di un risultato numerico o poco più. Una certa grafica primordiale si poteva ottenere stampando su un foglio una combinazione di simboli e caratteri che guardati da lontano potevano in qualche misura rendere l'idea di una figura bidimensionale.

Una Teletype o nella traduzione italiana una "telescrivente".



Il calcolatore non gode in principio di una periferica espressamente progettata per le sue esigenze. All'inizio infatti è stata la teletype a farla da padrona. Come spesso succede infatti, le nuove tecnologie utilizzano in parte quello che si trova sul mercato in quel momento e la telescrivente esisteva nelle aziende (di solito una) ma soprattutto esistevano i fornitori che le costruivano. Immaginiamo i salti di gioia di codesti imprenditori che si videro moltiplicare le possibilità di vendere i loro prodotti, eventual-

mente un poco adattati, nel settore emergente del Corporate Computing.

Il terminale all-in-one video-tastiera ha conosciuto molteplici implementazioni e una discreta fortuna. Ad esempio la nostrana Olivetti ne ha cavalcato il successo producendo sistemi di videoscrittura come le famose BCS (che sinceramente non mi sovviene di quale combinazione di parole sia la sigla, qualcosa come Business Calcolator System o magari qualcosa di molto più italico...).



Lo scopo di questo articolo non è l'evoluzione tecnologica dell'hardware, per cui non parleremo di come la tastiera si sia evoluta e abbia coinvolto tutte le possibilità elettro-meccaniche, chimiche, ottiche,… possibili ed immaginabili per ottenere dei tasti da pigiare (magari solo virtualmente).

Dal punto di vista dell'interazione uomo-macchina, la tastiera ha cercato di venire incontro alle molteplici esigenze che i nuovi sistemi operativi e soprattutto programmi applicativi si sono inventati nel corso del tempo. Ci ha pensato Apple per prima a stabilire che era necessario il "tasto mela", dato che il classico "Control" già peraltro presente nelle tastiere industriali, non era sufficiente.

I tasti funzione sono una invenzione che risale all'epoca dei terminali TTY dei mainframe. Ancora oggi nei file di configurazione dei vari Unix (ad esempio Linux) si parla di tasti PF1…PFn, che è il nome che dava IBM ai tasti "fuori testo" nei propri terminali.

Una evoluzione in linea con questo è stata la comparsa di tasti legati a specifici programmi e più in generale ai tasti programmabili, cioè dove l'utente può intervenire per associare una sequenza di comandi allo stato di "tasto premuto" rilevato dalle routine di input del sistema operativo in uso (normalmente sono pre-programmati solo per Windows). L'idea non è comunque nuova e già i costruttori dei primi home a volte la prevedevano come features per promuovere l'usabilità delle proprie realiz-

Olivetti BCS. Quella mostrata in fotografia (la macchina, non la ragazza!) è in realtà una versione molto evoluta con tanto di doppia unità floppy da 8 pollici.



La tastiera del terminale 3270 della IBM. I tasti funzione PFxx sono quelli in doppia fila sulla sinistra.

zazioni.

La tastiera, oltre a diventare più tecnologica (senza fili ad esempio), ha provato ad integrare altre periferiche di input: il mouse prima di tutto. Si tratta di poco più che esperimenti, peraltro non nuovissimi: personalmente possiedo una tastiera della Olivetti collegabile al PC attraverso infrarossi e che ingloba una piccola trackball.

Concludendo questo excursus sulla periferica di input per eccellenza, possiamo senz'altro affermare che la tastiera ha attraversato tutta l'epopea della conquista dell'informatica arrivando indenne, anzi ringiovanita, al traguardo dell'era di Internet.

Il mouse

L'avvento del mouse ha segnato uno snodo epocale nella storia delle interfacce utente. Non è che questo dispositivo di input sia sorto dal deserto di idee precedente, anzi! Più o meno genialmente altri dispositivi di input sono stati progettati e inseriti in mercati verticali, ad esempio nella grafica tecnica, con puntatori più o meno precisi per la rilevazione delle coordinate ad esempio geografiche o semplicemente cartesiane.

Ricordo a questo proposito una geniale applicazione di un plotter HP collegato all'HP85, che permetteva anche l'input. Funzionava più o meno così: si metteva il grafico con la curca da rilevare sul piano del plotter e si faceva partire il programma che rilevava l'input. A questo punto era l'abilità dell'operatore nello spostare il pennino per posizionarlo esattamente sulla curva operando con i tasti di movimento manuale cui la periferica era dotata. Ad ogni punto un "invio" indicava al computer di registrare la posizione del pennino, cioè le coordinate cartesiane del plot della curva da rilevare. Questa applicazione era resa possibile per il fatto che l'interfaccia di collegamento PC-Plotter, era la HP-IB, una specie di seriale a bus bidirezionale.

Il mouse stesso ha vissuto diverse fasi e adattamenti in relazione al peso specifico di questa metodo di interazione rispetto all'interfaccia utente studiata per un applicativo o più in generale per il sistema operativo della macchina.

Pallina rotolante, rilevatori ottici, altre diavolerie (fino al laser), hanno visto la loro applicazione più o meno riuscita al mouse, per non parlare degli innumerevoli esercizi di design, tutti fatti passare per

"ultimate", che avevano lo scopo dichiarato di rendere confortevole l'uso di questo strumento.

Contrariamente a quanto molti credono, il mouse sbarca nei personal computer quando ancora l'interfaccia grafica non era una realtà. Fra il terminale a caratteri e Windows (tanto per capirci), c'è una "terra di nessuno" nella quale sul video si sono accomodate finestre delimitare da simboli semigrafici entro le quali il mouse aveva ragione di esistere.

Uno dei primi personal a consentire l'uso del mouse è stato il venerabile Apple IIe/IIc che fece propria la periferica studiata per la nuova macchina di Apple, cioè il Mac. I PC propriamente detti, cioè le macchine IBM compatibili o DOS Machine, hanno anch'esse tentato di ricavare vantaggio da quella piccola appendice rotolante che tutto sommato costava poco e rendeva il sistema più moderno.

Sono stati i sistemi alternativi al DOS o estensioni dello stesso, i primi ambienti che hanno pensato all'uso del mouse come utile strumento di interazione con l'utente. Deskview è un esempio di questa tendenza, ma anche programmi nativi per il sistema operativo di Microsoft ne hanno cavalcato l'onda. Uno che mi viene in mente, perché l'ho usato parecchio, è il Turbo Pascal della Borland che nelle sue release più avanzate (ad esempio la 7), ha fatto un uso sistematico del concetto di punta e clicca.

[Tn]



[Sopra]
La versione Apple del mouse per il suo Mac: un solo tasto e una ergonomicità che oggi sarebbe inaccettabile.

[Sotto]
lo Smaky Mouse, inventato al Politecnico di Losanna; a fianco una versione del computer Smaky con il relativo mouse collegato.





Biblioteca

Le monografie vecchie e nuove che rappresentano una preziosa risorsa per chi ama il mondo dei computer in generale.

Scheda

Titolo:

After the Software War

Autore:

Keith Curtis

Editore:

Keith Curtis

Anno: 2009

Lingua: Inglese

ISBN:

978-0-578-01189-9

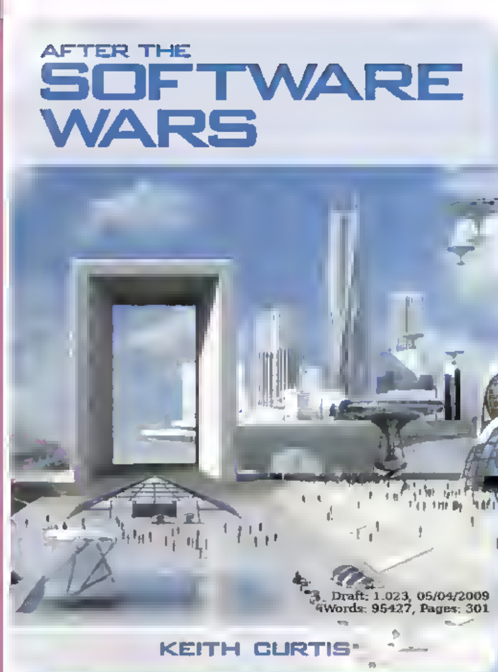
Note:

Free download

pagine:

301

After the Software War



Incomincia con questa citazione (del 2004) di John McCarthy, uno dei padri indiscussi della disciplina che va sotto il nome di Intelligenza Artificiale:

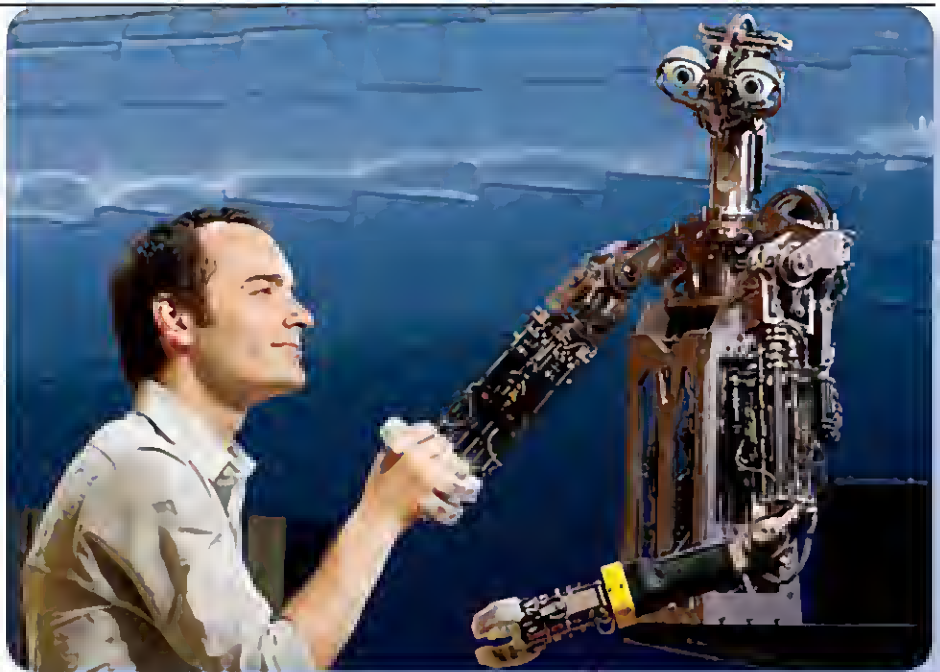
"Some people think much faster computers are required for Artificial Intelligence, as well as new ideas. My own opinion is that the computers of 30 years ago were fast enough if only we knew how to program them."

Significa terra-terra che il software è la vera macchina e che l'hardware è solo un supporto, purtroppo sempre sotto-utilizzato rispetto alle potenzialità che il buon software potrebbe abilitare (e nobilitare) in esso.

La tesi che sostiene l'autore è la seguente: -"Se i buoni cervelli impegnati allo sviluppo del software smettessero di occuparsi della guerra compagnia contro compagnia, avremmo ora un software molto ma molto più avanzato rispetto a quello che possiamo utilizzare ora".

Una tesi interessante e lapalissiana, oserei dire. E' ben evidente infatti che se tutti, ma proprio tutti, fossero impegnati a sviluppare un unico sistema operativo e un'unico software gestionale, office, etc..., la massa critica "grigia", non potrebbe che generare un compendio più avanzato di quanto usiamo giornalmente. E' anche vero, per essere obiettivi, che lo sviluppo di certi comparti (penso alla interfaccia utente piuttosto che alle prestazioni grafiche) sono state rese possibili dai massicci investimenti proprietari. E, voi mi insegnate, una azienda investe se ha (o ha speranza di avere) un ritorno economico dal prodotto che sviluppa.

Se ci si ferma a pensare a quanti software ci sono in giro che fanno grossomodo la stessa funzione, ci si rende conto che John ha azzeccato il succo della faccenda. Non stimao pensando a Linux verso Windows o a MSOffice verso OpenOffice o ancora SQLServer verso



Oracle, stiamo pensando alla miriade di applicazioni sviluppate da "solisti" e che si sovrappongono una sull'altra spesso con differenze minime.

Tempo fa stavo cercando un software per il time-tracking, che per chi non lo sapesse è un qualcosa che aiuta a tenere traccia di quanto tempo si passa su un progetto. Pur limitando la ricerca al sistem operativo MAC OSX, l'unico che uso per lavoro, mi sono trovato difronte a qualche decina di possibilità fra le quali scegliere. Bene, analizzandone qualcuna le differenze erano veramente minimali, soprattutto limitandosi a confrontare le funzionalità piuttosto che l'interfaccia, i colori, l'amichevolezza, etc...

Il volume di 300 pagine, pubblicato secondo gli standard della filosofia Creative Commons, è un buon testo per capire lo stato dell'arte del software nei vari comparti e la storia e retroscena che ne ha caratterizzato lo sviluppo.

Conclusione.

E' molto intrigante l'idea di un mondo possibile che avrebbe potuto svilupparsi se l'industria informatica avesse seguito strade diverse rispetto a quella protezionistica (il male in fondo deriva da quella prima decisione).

E' anche vero che, come si dice "il modo è bello perché è vario" e alla resa dei fatti l'attuale situazione che vede molti concorrenti e molti prodotti che fanno più o meno la stessa funzione, è comunque un "buon mondo", basta pensare a dove eravamo appena venti anni orsono (cioè al DOS di Microsoft) per convincersene.

[Sn]

Una delle immagini finali del volume; riassume bene l'idea di evoluzione del software rispetto alle potenzialità finora non espresse.
(Sotto) In questa raccolta di simboli (tratta dal volume), sono rappresentate le features (alcune palesemente provocatorie) che potremmo avere ma che non abbiamo ancora.



Retro Riviste

La rassegna critica dell'editoria specializzata dai primi anni '80 ad oggi

Super Apple



Negli anni dal '84 al '90 circa abbiamo assistito al fenomeno dell' home computer e con esso alla corrispondente nascita di numerose testate mono-marca se non addirittura mono-modello.

Piccoli calcolatori personali, pochissimo equipaggiati in termini di memoria e unità magnetiche, venivano spacciati per super macchine in grado di svolgere compiti non solo domestici domestici ma soprattutto giochi e agende personali. Non mancava qualche speranzoso operatore pubblicitario che cercava di vendere il tutto alla massaia di turno con la scusa delle ricette, della lista della spesa e del calcolo del detersivo per il lavaggio settimanale.

Capostipiti di questa invasione furono i computer della Commodore (Vic, 64), della Sinclair (ZX81, Spectrum) e Apple (IIe, IIc). Ma numerose altre case costruttrici misero sul mercato i loro modelli cercando di ritagliarsi una loro fetta di mercato. Molto interessante sarà il fenomeno degli "MSX", cioè macchine costruite da produttori diversi ma compatibili a livello di codice.

Il boom degli "home" fu una vera manna per i produttori: alta percentuale di ricarico, nessuna assistenza software e scarsa assistenza hardware; insomma un autentico paradiso anche se la sua durata è stata limitata a meno di un decennio. La manna finì nel momento in cui entrarono i cloni dei pc a prezzo "umano". Si pensi ad esempio che nel 1987 l'Olivetti M24 costava la bellezza di 7 milioni e l'originale IBM più di 10! Era logico che l'utente non discendente dal pascià di Persia, si accontentasse delle macchine da 500/600 mila lire.

Gli home nascevano con un loro sistema operativo più o meno sofisticato ma con una dotazione di software scarsissima. Ciascuno

Scheda

Titolo:
Super Apple

Sottotitolo:
La rivista per gli utenti dei prodotti Apple

Editore:
J.Soft

Lingua:
Italiano

Primo numero:
1985

Prezzo
L. 5.000

però poteva creare i propri programmini grazie all'onnipresente interprete Basic.

Ecco quindi l'utilità delle riviste super-specializzate, ricche di listati e, qualche volta con un floppy o una cassetta. Chi acquistava un certo modello era poi praticamente obbligato a comprare la rivista di riferimento, per il software senza dubbio, ma anche semplicemente per sentirsi meno solo. A riprova di queste affermazione la nascita coeva di numerosissimi "Computer Club", spesso editori di "fanzine" dalla bassa tiratura e dalla periodicità del tutto casuale, ma dai contenuti quasi sempre interessanti.

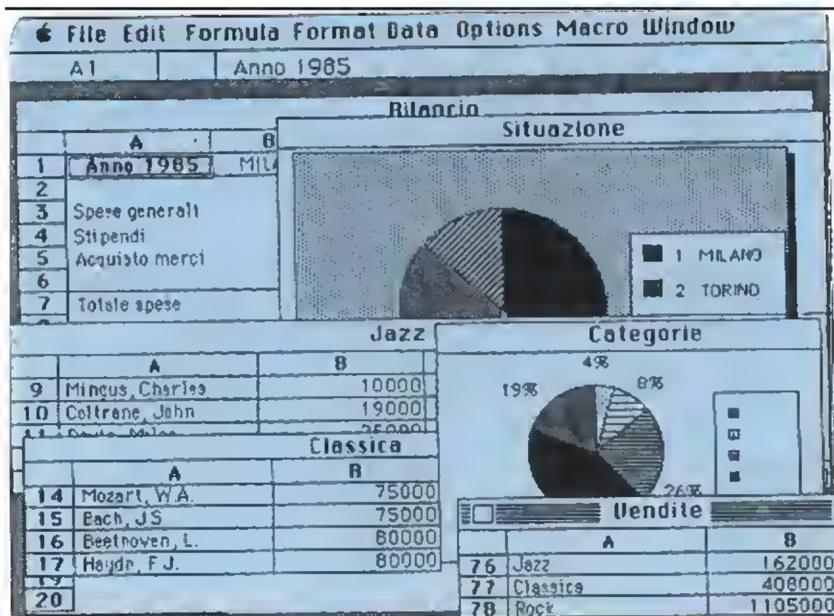
"Super Apple" è una rivista monomarca, interamente dedicata al software per i sistemi Apple, edita dal gruppo editoriale Jackson. Lo stuff editoriale è interamente italiano e questo è un segnale importante: l'editore non ha seguito la più facile strada della "rivista tradotta".

Vale a questo punto la pena che io spenda due parole riguardo proprio all'editore. Il gruppo editoriale Jackson mi ha ben presto deluso: ha prodotto moltissimo ma quasi sempre prodotti "di cassetta" pronto ad abbandonare la barca al primo calo di vendite. Le traduzioni dei libri poi sono state a volte degli autentici "scapolavori", della serie "lo speriamo che me la cavo". Cio' nonostante è stato un editore importantissimo per la diffusione dell'informatica personale in Italia e di cio' gli si deve dare atto.

Evidentemente il mondo Apple dispone di un installato nostrano di numerosità consistentente, tale da convincere qualcuno nella bontà dell'impresa. La diffusione Apple in Italia non è niente di paragonabile a quella degli Stati Uniti, per merito di una diversa scala reddituale (il loro potere d'acquisto è sempre stato di gran lunga superiore al nostro), ma anche di una lungimirante politica di diffusione nelle scuole e università. Il marketing della Apple di allora era chiarissimo: "Se lo studente trova il nostro computer nella scuola ed impara su di esso, poi quando sarà in azienda spingerà che venga adottato anche lì".

La parte "Mac" della rivista è nettamente divisa dalla parte "II" e confinata nella zona denominata





"SuperMac". Il taglio è decisamente meno tecnico rispetto alla controparte dedicata ai "piccoli" della serie II. Incominciano ad imporsi applicativi da ufficio della serie "office" e quindi prove di Word e fogli di calcolo in Excel hanno ampio risalto. Certo che a vederle ora queste schermate di Excel fanno sorridere, ma il foglio di calcolo della Microsoft sul Mac è stata una autentica rivoluzione nel campo della usabilità.

Dedicato ad "AppleWorks", con l'analisi del modulo "foglio di calcolo", un lungo articolo nella rubrica "Profilo Software" nella parte di rivista dedicata all'Apple II. AppleWorks è stato il padre di tutti i software "suite", cioè delle raccolte di programmi collaborativi che possono scambiarsi i dati. L'idea era rivoluzionaria per l'epoca ma il "copia-incolla" diventerà l'indispensabile feature di ogni integrato degno di questo nome (come potremmo farne senza?).

Enrico Colombini si dimostra un vero esperto della mela e un buon terzo dell'intera rivista nasce dalla sua penna (il che a volte lascia abbastanza sconcertati: possibile non ci siano buoni giornalisti tecnici in grado di scrivere sui prodotti della Apple Computer?). Si va dal corso di Assembly per il 6502 alla presentazione di "Melopoli", il Monopoli programmato in Basic dallo stesso autore dell'articolo.

Non manca l'articolo divulgativo: "Banche dati". Il modem comincia a suscitare interesse; nascono le prime "banche dati", anche se per la nascita delle fenomeno "BBS" c'è ancora tempo.

Super Apple Programmi

Come tutte le riviste anche Super Apple arricchisce i propri servizi verso il lettore con una ricca raccolta mensile di programmi pronti per essere digitati e, quindi, eseguiti sul proprio Apple IIe (o altro modello compatibile).

Simpaticamente il sommario viene chiamato "CATALOG" che è appunto il comando del DOS (quello Apple, non quello della Microsoft per PC J). Vi troviamo la solita lista che spazia dai giochini tipo "l'impiccato" alle utilità tipo "tastiera magica".

Non mancano i gestionali: "archivio Apple" ad esempio è un programma di circa 300 righe che permette di tenere un vero e proprio archivio del proprio materiale informatico. Niente a che fare con

dBase o altro più famoso, naturalmente. Semplicemente un piccolo programmino dal quale trarre utili spunti di programmazione.

Negli anni '80 le riviste di informatica erano abbastanza poco approfondite, mancava Internet e la letteratura tecnica scarseggiava alquanto. I sorgenti erano una fonte insostituibile di trucchi e tecniche di programmazione introvabili altrove; la semplicità del Basic faceva il resto.

Il sorgente più tosto contenuto in questo fascicolo è sicuramente quello chiamato "Window" che ha come scopo dimostrare l'utilizzo di due funzioni davvero incredibili che permettono di creare delle finestre un po' come sul Mac Intosh (Windows non è nemmeno nella mente di Bill Gates). Le due funzioni si realizzano semplicemente con due CALL e le loro istruzioni sono, manco a dirlo, farcite di PEEK e POKE che salvano e ripristinano porzioni di video.

E il mouse? Beh, state pretendendo un po' troppo. Il Mouse per Apple IIe esiste ma l'uso è tutt'altro che banale. Credo che sarà necessario attendere il ProDOS per vedere all'opera qualcosa che assomigli ad una interfaccia a finestre con tanto di gestione dal mouse.

Conclusione

La strada dell'informatica è irta di molti pericoli per i sistemi della mela e non solo dovuti alla comparsa dei cloni IBM. Scelte di marketing a dir poco discutibili, disaffezione degli utenti e migrazione verso il meno invitante ma più pratico "mondo IBM", lasceranno la Apple Computer con poco più di un pugno di mosche in mano. A dispetto delle funzionalità limitate dei primi PC (intendendo con la sigla "PC" le macchine equipaggiate con i processori Intel ed architettura IBM) come ad esempio nella grafica per non parlare del suono, pur tuttavia la strada è inesorabilmente tracciata e saranno le "DOS Machine" a dominare il mercato di qui a pochi anni.

[Sn]

La copertina del fascicoletto allegato "Super Apple Programmi".

TAMC

Algoritmi di SORT (parte 5)

Teoria e Applicazioni delle Macchine Calcolatrici: la matematica e l'informatica, le formule e gli algoritmi, la completezza e la computabilità, le strutture dati e tutto quello che sta alla base dell'informatica.

Heap Sort

L'Heapsort è un algoritmo di ordinamento che si basa su una struttura computazionale chiamata heap. Dal punto di vista teorico un heap è una struttura dati che riceve dei valori e li mantiene ordinati facendo "emergere" come primo elemento quello che ha una particolare caratteristica (ad esempio la priorità) più alta degli altri elementi.

Una maniera pratica di realizzare questo è disporre di un vettore nel quale gli elementi sono mantenuti ordinati da un qualche algoritmo che già conosciamo, ad esempio un Quicksort, cosa che ovviamente non aggiunge nulla di nuovo all'efficienza per ottenere un ordinamento di elementi, oltre al fatto che realizzare una struttura heap procedendo all'ordinamento completo di un vettore ogni volta che viene inserito un elemento, è a dir poco inefficiente. Pertanto sono stati studiati metodi migliori per realizzare ciò.

Una prima osservazione è che un heap può essere rappresentato da un albero binario. Vediamo in figura 1 come sarebbe costruito l'albero per il vettore di esempio:

[3, 5, 7, 1, 4, 6, 2]

Si vede come questa rappresentazione permetta di sapere qual'è l'elemento maggiore dell'insieme, visto che è la radice stessa dell'albero.

Estratto il primo elemento (il valore 7 nel caso di esempio), l'albero va ricostruito (Figura 2).

Notate due cose che sono evidenziate dalle figure: l'albero non è ordinato ma semplicemente si assicura che la radice di un nodo sia maggiore dei suoi figli e seconda osservazione: l'albero è completo, cioè i nodi finali (foglie) sono tutti portati allo stesso livello di profondità.

Questa seconda caratteristica (la completezza) non è strettamente essenziale se si volesse usare l'albero come struttura dati per lo heap. In realtà l'albero binario completo ha un vantaggio computazionale rispetto allo stesso albero non completo: può essere rappresentato in memoria come una matrice.

Nello studio delle strutture dati gli alberi sono costruiti solitamente con due elementi: i nodi e i link. Il nodo contiene il valore in esso

posizionato e poi n link (due nel caso di albero binario) che puntano ai nodi figli.

Il problema di questa struttura computazionale è che la ricostruzione dell'albero nel caso di estrazione di un nodo, diventa onerosa; in pratica andrebbe ricostruito l'intero albero. Ovviamente se vogliamo guadagnare qualche cosa nei tempi di ordinamento dobbiamo sfruttare qualcos'altro e questo altro è appunto la rappresentazione dello heap come vettore di n elementi (n è anche la dimensione del vettore da ordinare).

Se il nodo radice occupa la prima posizione del vettore, i suoi figli sono rispettivamente nella seconda e terza posizione:

[7, 5, 6,...]

i successori del nodo [5] vengono inseriti in quarta e quinta posizione:

[7, 5, 6, 3, 1,...]

insomma il successore sinistro del nodo i si trova nella posizione 2i dello heap, mentre il successore sinistro si trova in 2i+1.

La regola è quindi molto semplice e così pure dato il nodo in posizione i, sappiamo che il suo genitore si trova in i/2.

Trattare un albero descritto in questa maniera è, lo si capisce perfettamente, un grande vantaggio in termini computazionali ed è proprio qui che heapsort guadagna.

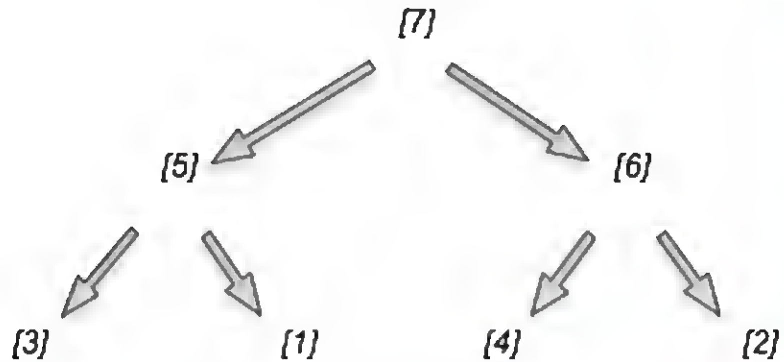


Figura 1. L'albero binario iniziale costruito con gli elementi del vettore da ordinare.

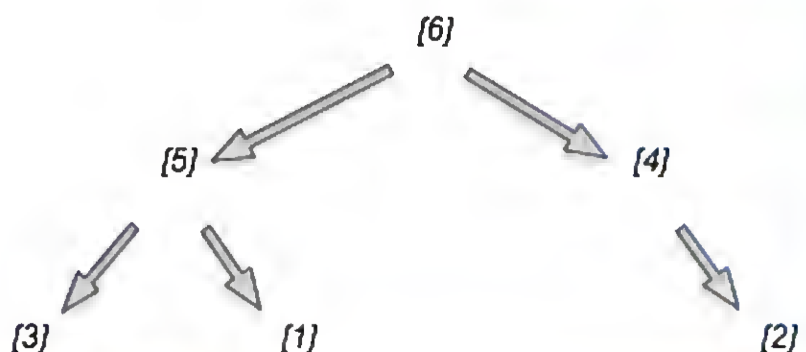
Nel listato pubblicato nella pagina seguente, vediamo la semplicità dell'algoritmo.

Il programma è ridotto a due loop lineari; il primo inserisce gli elementi nello heap e il secondo li estrae.

Le due funzioni che sottostanno alle due attività le abbimao chiamate **putheap()** e **getheap()** rispettivamente.

L'aggiunta di un nodo allo heap consiste nel far occupare al nuovo elemento la prima posizione libera e poi "migrarlo" eventualmente verso l'alto del vettore confrontandolo con il suo genitore ed eventualmente scambiandolo con esso. Si tratta quindi di un semplice confronto con log(n) elementi che va

Figura 2. La prima estrazione toglie l'elemento più alto dallo heap e riorganizza l'albero.



```
Program sortheap;

var
A, B : array[1..7] of integer;
i : integer;

procedure putheap(i: integer);
var
k, P : integer;
temp : integer;

begin
    k := i;
    p := k div 2;
    B[k] := A[k];
    while (k > 1) and (B[k] > B[p]) do begin
      temp := B[p];
      B[p] := B[k];
      B[k] := temp;
      k := p;
      end;

    writeln(B[1], ' ', B[2], ' ', B[3], ' ', B[4], ' ', B[5], ' ', B[6], ' ', B[7]);
end;

function getheap(n: integer): integer;
var k, p : integer;
   temp : integer;

begin
    getheap := B[1];
    B[1] := B[n];
    B[n] := 0;
    k := 1;
    while (k < n) and ( (B[k] < B[2*k]) or (B[k] < B[2*k + 1])) do begin
      if (B[2*k] > B[2*k + 1]) then begin
        temp := B[k];
        B[k] := B[2*k];
        B[2*k] := temp;
        k := 2*k;
        end
      else begin
        temp := B[k];
        B[k] := B[2*k + 1];
        B[2*k + 1] := temp;
        k := 2*k + 1;
        end;
      end;
  writeln(B[1], ' ', B[2], ' ', B[3], ' ', B[4], ' ', B[5], ' ', B[6], ' ', B[7]);

end;

begin
  A[1] := 3;
  A[2] := 5;
  A[3] := 7;
  A[4] := 1;
  A[5] := 4;
  A[6] := 6;
  A[7] := 2;

  writeln('Heap Sort');
  writeln;
  writeln;
  writeln(A[1], ' ', A[2], ' ', A[3], ' ', A[4], ' ', A[5], ' ', A[6], ' ', A[7]);
  writeln ('===============');
  for i:= 1 to 7 do putheap(i);
  writeln ('===============');
  for i := 7 downto 1 do A[i] := getheap(i);
  writeln ('===============');
  writeln(A[1], ' ', A[2], ' ', A[3], ' ', A[4], ' ', A[5], ' ', A[6], ' ', A[7]);
  readln;

end.
```

eseguito per gli n elementi del vettore.

L'operazione non bilancia l'albero binario in senso stretto, cioè è possibile che un figlio risulti maggiore del rispettivo padre per effetto dell'inserimento di un nodo che ha sostituito il genitore.

Per quanto un albero binario bilanciato sarebbe migliore per procedere all'ordinamento, l'operazione viene rimandata al momento dell'estrazione del nodo radice che è per destinazione l'elemento più grande dell'insieme da ordinare. Si noti che l'operazione di parziale ordinamento che viene messa in atto dalla funzione getheap, viene eseguita su un albero ridotto di una dimensione in profondità (ha perso la radice), il che rende ancora più efficiente l'algoritmo.

*Come nel caso del Mergesort, anche l'Heap sort ha un comportamento costante rispetto alla complessità di esecuzione nei casi che si possono presentare. Infatti l'ordine è **$O(n\log_2 n)$** per tutti i casi.*

Abbiamo così esaminato due algoritmi (Heapsort e Mergesort) che si dimostrano comparabili dal punto di vista della complessità, e danno tempi di esecuzione migliori rispetto agli algoritmi esaminati in precedenza, soprattutto nel caso peggiore. Possiamo inserire fra di essi anche Quicksort che è penalizzato solo dai casi sfortunati (è dell'ordine di n^2).

La domanda è: -"Quale dobbiamo scegliere?"

Ovviamente dipende dalle circostanze; infatti non è detto che non si conosca proprio nulla del sistema da ordinare. Ad esempio può essere noto che il vettore è parzialmente ordinato o magari ordinato quasi del tutto e in questo caso la scelta Quicksort è migliore dei concorrenti. Questo caso succede quando si debba inserire un insieme di numeri piccolo rispetto all'insieme originale già ordinato. La strategia è quella di aggiungere semplicemente gli elementi nuovi in testa o in coda all'insieme originale e poi lanciare su di esso Quicksort che troverà il vettore già ordinato o quasi.

Mergesort e Heapsort hanno uno svantaggio rispetto a Quicksort: la dimensione di memoria che occupano è il doppio dell'insieme da ordinare. Questo perché richiedono delle strutture dati di appoggio che hanno dimensione N, se N è la dimensione dell'insieme da ordinare. Peraltro se stiamo ordinando oggetti diversi da semplici numeri interi, mergesort potrebbe essere la scelta migliore perché permette la divisione fisica di sottoinsiemi che si possono anche "appoggiare" su un file-system (è il caso di interi record all'interno di un file).

Heapsort è una soluzione "di nicchia", la sua utilità si evidenzia in due casi: quando il linguaggio che usiamo non supporta la ricorsione o è penalizzato dall'utilizzo delle chiamate rientranti (un esempio è il COBOL), oppure non serve specificatamente un ordinamento ma una estrazione da un insieme dell'elemento che ha una certa proprietà maggiore degli altri elementi. Questo è proprio il significato di uno heap: accettare valori e restituire sempre l'elemento maggiore senza operare un vero ordinamento dell'insieme residuo.

```
Heap Sort

3 5 7 1 4 6 2
===============
3 0 0 0 0 0 0
5 3 0 0 0 0 0
7 3 5 0 0 0 0
7 3 5 1 0 0 0
7 4 5 1 3 0 0
7 4 6 1 3 5 0
7 4 6 1 3 5 2
===============
6 4 5 1 3 2 0
5 4 2 1 3 0 0
4 3 2 1 0 0 0
3 1 2 0 0 0 0
2 1 0 0 0 0 0
1 0 0 0 0 0 0
0 0 0 0 0 0 0
===============
1 2 3 4 5 6 7
```

Figura 3. L'output del programma sul vettore di esempio..

Archiviato Heapsort stiamo volgendo verso la fine dell'esame dei sistemi di ordinamento "fondamentali" che sono studiati in algoritmica. La prossima volta vedremo un algoritmo che ha una complessità N anche nei casi peggiori: praticamente un sogno...

[Sm]

Retro Linguaggi

LISP (parte 2)

La storia dell'informatica è stata anche la storia dei linguaggi di programmazione.



Funzioni e liste

Dunque il LISP è un linguaggio orientato alla manipolazione delle liste, fino a qui lo abbiamo capito tutti. Ora da questa dichiarazione di intenti ci si aspetterebbe una lista infinita di funzioni che manipolano liste e invece, sorpresa: le funzioni sono sorprendente poche. Come mai?

La risposta è che qualsiasi funzione necessiti basta costruirsela! Bel proposito, per un programmatore moderno! Già, questi programmatori "moderni" che hanno la pappa precotta già belle e confezionata e al massimo ereditano una proprietà da una classe e ne definizcono due metodi in croce. Non che ci sia niente di male in questo, anzi! Solo che non è proprio in sintonia rispetto a come mi pongo difronte ad un algoritmo da implementare.

Bando alle ciance e veniamo ad esaminare le due funzioni cardine dell'intera programmazione LISP. Queste si chiamano CAR e CDR (pronuncia "car" e "cudar"). La prima restituisce il primo elemento della lista passata come argomento, la seconda restituisce la lista "decapitata" del primo elemento.

Esempio:

```
(car '(a b c)) -> a

(cdr '(a b c)) -> (b c)
```

semplice ed elegante (come sono sempre le cose semplici).

Se ci pensate con queste due funzioni se ne costruisce un bel po'...

Già, ma come si fa a definire una nuova funzione LISP? C'è a questo scopo una funzione speciale il cui compito è appunto costruire altre funzioni. Ecco la sua sintassi:

```
(defun name (parameter*)
       body-form*)
```

Esempio:

```
(defun somma (a b) (add a b))
```

Abbiamo definito una nuova funzione che si chiama "somma" e che si aspetta due parametri, qui indicati formalmente con le due lettere a e b. Il "corpo" della funzione è la sequenza di altre funzioni da chiamare e il risultato è esatta-

mente del tipo restituito dall'ultima funzione chiamata che esce dallo stack.

Definiamo ora una funzione che restituisce il secondo elemento di una lista.

```
(defun secondo (lista)
      (car (cdr lista))
)
```

Cioè: per ottenere il secondo elemento della lista "lista" basta prendere il primo elemento della lista decapitata del primo elemento.

Per esercizio provatevi a definire la funzione che restituisce il terzo elemento della lista. Non è complicato, ma se non ci riuscite ecco qua la soluzione:

```
(defun terzo (lista)
      (car (cdr (cdr lista)
               )
           )
)
```

Abbiamo indentato il codice e portato le parentesi di chiusura su righe separate per chiarezza espositiva. In realtà nel codice che si leggerebbe in giro il tutto è scritto "plan" senza grande importanza alla sintassi:

```
(defun terzo (lista)
      (car (cdr (cdr lista))))
```

Come si vede dall'essempio scritto, la proliferazione delle parentesi rischia di confondere la vista. Alcuni dialetti accetterebbero una sintassi con parentesi quadra a chiusura:

```
(defun terzo (lista)
      (car (cdr (cdr lista]
```

ma secondo "i puristi" è brutto da vedere e non così chiaro da leggere. Gusti sono gusti...

La valutazione del risultato

Come è ben noto a tutti (credo), quando si scrive codice si scrivono anche errori, effettivi o potenziali, cioè i famosi bug.

Ad esempio se applichiamo la funzione "secondo" ad una lista che ha un solo elemento cosa succede?

(secondo '(a))

da un risultato o da errore?

Cercando di ragionare come il sistema durante la valutazione, vediamo che prima di tutto viene risolta la funzione (cdr lista) che risponde con un NIL, cioè il puntatore vuoto perché in una lista di un solo elemento non c'è nessun cudar!

Poi questo risultato viene preso in carico dalla funzione car e la valutazione finale diventa:

(car NIL)

il cui risultato è NIL, cioè un puntatore vuoto.

Definizione di nuove funzioni

Si vuole costruire una funzione che data una lista ne restituisca un'altra formata solo dai primi due elementi della lista originale.

Ecco il codice:

```
(defun primi_due (lista)
   (list (car lista)
      (car (cdr lista))
      )
)
```

Riteniamo superfluo commentare questo codice che dovrebbe ormai esservi semplice da comprendere.

Ora la funzione applicata ad una lista:

(primi_due '(a b c d))

-> (a b)

Come si diceva la costruzione di una elaborazione LISP consiste nella definizione di funzioni che alla fine portano al risultato. Ad esempio per costruire una funzione che restituisca una lista dei primi tre elementi di un'altra lista sorgente potremmo scrivere:

```
(defun primi_tre (lista)
    (list (car lista)
      (car (cdr lista))
       (car (cdr (cdr lista)))
     )
)
```

Oppure, usando le funzioni primi_due e terzo, prima definite:

```
(defun primi_tre(lista)
   (list (primi_due(lista))
      (terzo lista)
   )
)
```

In effetti il codice appena scritto non restituisce esattamente il risultato desiderato.

Vediamo:

(primi_tre '(a b c d e f))

-> ((a b) c)

cioè non si tratta propriamente di una lista con i primi tre elementi ma di una costruzione più complessa: una lista formata da una sottolista e dal terzo elemento della lista sorgente.

Per ottenere il risultato voluto dovremmo modificare la nostra definizione:

```
(defun primi_tre (lista)
 (list (car (primi_due lista))
 (car (cdr (primi_due lista)))
      (terzo lista)
 )
)
```

Nuove funzioni matematiche

Vediamo un esempio di definizione di una funzione che fa uso delle operazioni aritmetiche.

Supponiamo di voler definire una funzione che applicata a due numeri restituisca il resto della division. Chiameremo questa funzione RESTO, con uno slancio di fantasia :-)

Dato che la funzione DIV restituisce il quoziente della divisione, faremo uso di una funzione predefinita built-in che si chiama INT e che restituisce la parte intera di un numero.

Ad esempio:

(INT 3.14) -> 3

quindi ad esempio:

(DIV 10 3) -> 3.33

(INT (DIV 10 3)) -> 3

Questo è quello che ci serve per ricavare il resto, infatti scriveremo:

```
(defun resto (a b)
    (sub b
        (mult a
            (int (div a b))
         )
     )
)
```

Ora, basta provare:

(resto 10 3) -> 1

Il bello dei linguaggi funzionali è che la costruzione di nuove funzioni è un processo del tutto naturale e supportato in maniera nativa.

Il LISP non possiede di suo una libreria matematica particolarmente sofisticata. Esistono le funzioni trascendenti e trigonometriche e poco altro, ma come abbiamo visto non è difficile procedere alla costruzione di una propria libreria di funzioni utili all'elaborazione che affrontiamo in quel momento.

Concludiamo questa seconda puntata del corso ricordando che il LISP è il linguaggio di programmazione interno ad Autocad, il software di disegno tecnico leader del mercato.

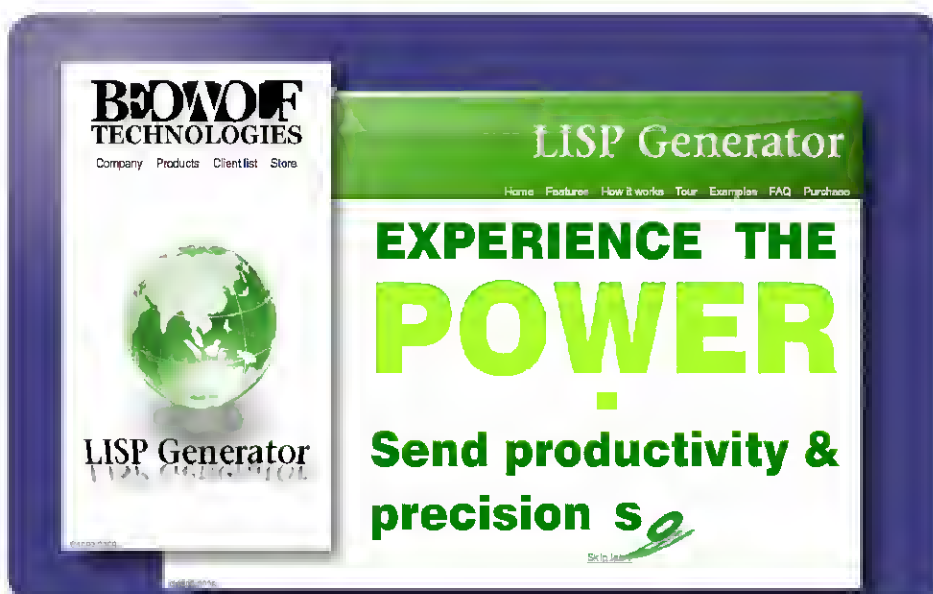
Tale implementazione viene chiamata Visual Lisp ed in rete si trovano numerose risorse ad esso dedicate.

[Sm]



The Visual LISP Developers Bible 2003 Edition (v2)
By David M. Stein

Visual LISP Development with AutoCAD 2004



Sopra: la copertina del libro "The Visual LISP Developer's Guide"

a fianco: il sito del prodotto "LISP Generator"

Edicola

In edicola o sul Web le riviste che parlano di computer, preferibilmente retro e free

Scheda

Titolo: *Abandoned Times*

Sottotitolo: *Abandoned but not forgotten*

Editore: *Red Avatar*

Web: *www.abandonia.com*

Lingua: *Inglese*

Prezzo: *Free*

Primo numero: *Maggio 2009*

Abandoned Times



Il retro gaming è sicuramente una branca notevole delle attività che riguardano il recupero dei vecchi calcolatori e in particolare il loro uso.

In effetti, se andiamo ad esaminare per cosa sono usati eventualmente le vecchie piattaforme di elaborazione, scopriremo che per la maggior parte dei casi si tratta di piattaforme adattate al gioco. Cioè si conserva il vecchio home per farci girare i vecchi titoli ludici ai quali siamo affezionati, mentre è solo marginale l'utilizzo con altri applicativi.

Se i vari programmi applicativi, ad esempio di produttività personale come word-processor o simili, hanno evoluto le loro prestazioni e in definitiva arricchita la loro offerta di servizi e di efficienza per l'utilizzatore, nel caso del gioco non è affatto così scontata l'evoluzione. Ci possono essere cioè dei giochi che sono inarrivabili per giocabilità e atmosfera anche se magari la grafica è blocchettosa, i colori sono solo quattro e il 3D manca del tutto.

Fra i titoli che gradualmente escono dai listini e dalla disponibilità del mercato ci sono anche migliaia di giochi che ogni anno vengono abbandonati al loro destino. La definizione di abandoned-game fa riferimento al software non più supportato e non più disponibile alla vendita per il quale chi detiene i diritti di sfruttamente ne ha liberato l'uso. Pur essendo una minima parte rispetto alla massa di titoli, i giochi "liberati" costituiscono comunque una numerosità notevole e un parterre interessantissimo.

Non mancano i siti Internet curati da appassionati del retro-gaming fra i quali Abandonia (indirizo www.abandonia.org) che è focalizzato principalmente (se non in maniera esclusiva) verso la piattaforma PC.

Recentemente gli amici del sito Abandonia, che fa parte del "giro" che regolarmente visito, hanno

dato vita ad una iniziativa editoriale con nome "Abandoned Times" che intende essere un po' quello che Jurassic News è per il retro computing in generale.

L'idea di costruire delle fanzine ad immagine di rivista periodica prende sempre più piede (in realtà non è mai morta questa idea), anche se poi, come sappiamo benissimo, queste iniziative raramente sopravvivono a qualche numero.

Speriamo che Abandoned Times possa proseguire e costituire un interessante archivio di recensioni, interviste, prove varie dei titoli che hanno appassionato moltissime persone negli anni dal 90 in poi.

La piattaforma PC, come tutti gli appassionati sanno, ha vissuto fra il 1990 e il 2000 un'epoca d'oro per quanto riguarda le uscite di titoli di intrattenimento. In quegli anni le console rimasero un po' al palo e i vecchi home mostrarono il fiato corto rispetto alle crescenti esigenze di coinvolgimento da parte del giocatore incallito, ma anche occasionale.

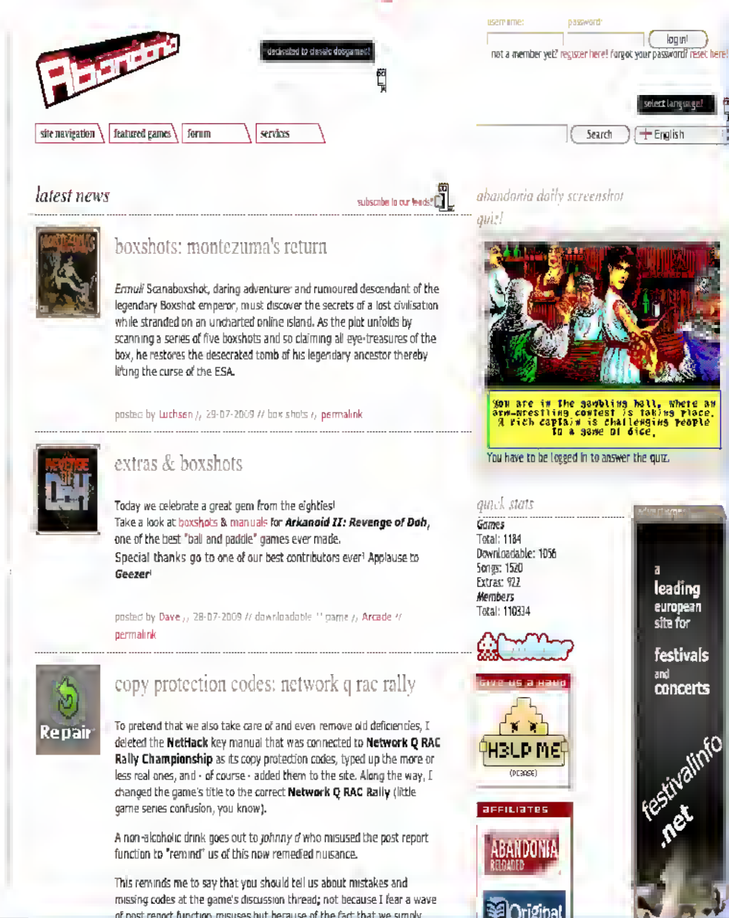
Ci sono stati dei titoli vere "killer applications" rispetto alla piattaforma. Mi viene in mente ad esempio "Flight Simulator" di Microsoft o "Doom", tanto per citare due giochi che tutti conoscono, ma ovviamente la lista sarebbe lunghissima.

Venendo alla fanzine, "Abandoned Times" per ora ha solo 12 pagine, coloratissime e curate dal punto di vista grafico, presenta un sommario organizzato in rubriche, che riporta:

- *Gaming Sins: how multi-platform releases hurt the PC games industry.*
- *Review: DreamWeb*
- *Listen Up! 2D or not 2D*
- *The Underground: Betasuppe*
- *Honour. Fallout VS Fallout 3*
- *Review: MechWarrior*
- *Loving Memory: Commander Keen*

In conclusione, una rivista completamente free che è piacevole da leggere e che rivela molte cose sui vecchi titoli per PC.

[L2]



La home page del sito Abandonia. Spartano ed essenziale.

SAP Corner

SAP NetWeaver 7.0 Trial Ed.

Una nuova rubrica dedicata ad un ambiente non retrò che ospitiamo in esclusiva per l'Italia

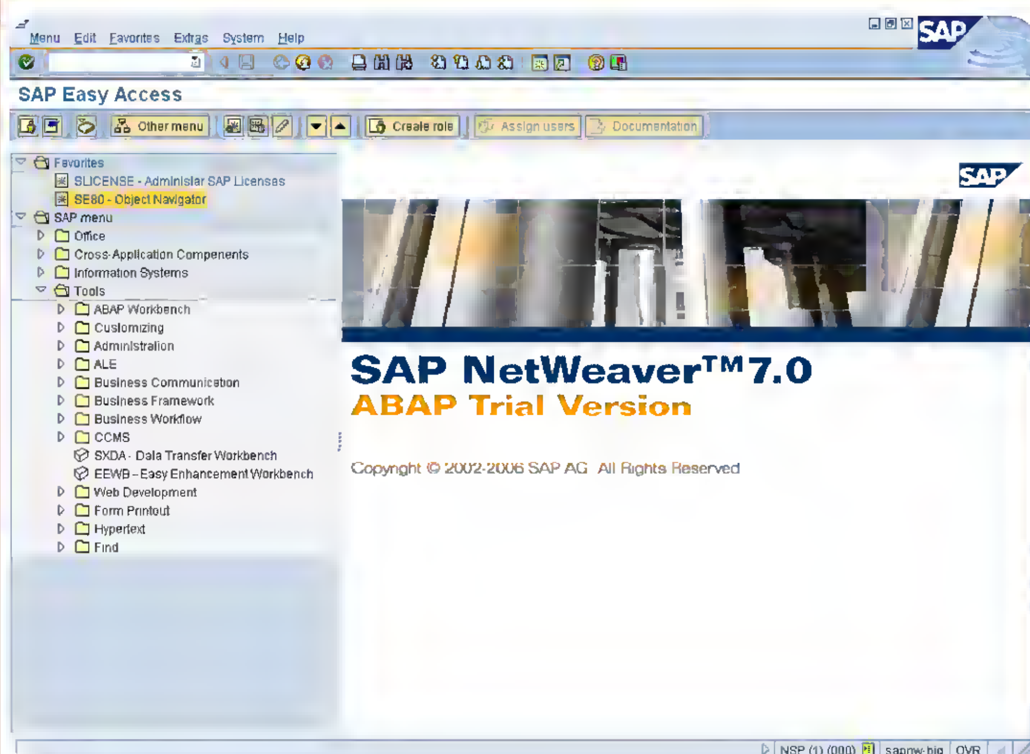


Figura 1. L'interfaccia.

Tutti i software si evolvono, così accade anche per l'ambiente ERP per eccellenza: cioè per SAP.

Dopo aver lavorato sulla "vecchia" 4.6, è venuto il momento di aggiornare la nostra installazione, tanto per tenersi allineati con le novità.

Abbiamo scaricato la nuova versione del trial dal sito "educational" di SAP via registrazione gratuita. I due file .RAR di circa 1,2 Gb cadauno vanno scompattati per avere un'unica directory di installazione contenente tutto l'occorrente (2,8 Gb circa).

L'installazione è simile a quella che abbiamo già intrapreso per la versione precedente: si parte con la pagina html che riporta le informazioni principali e il link per far partire effettivamente il programma di Setup (figura 2).

Per quanto non strettamente necessario, è bene disinstallare la precedente versione, altrimenti si potrebbero verificare delle interferenze nel registry della macchina (informazione comunque riportata in dettaglio nella documentazione di installazione).

Dal punto di vista tecnologico la versione 7.0 è molto diversa dalla 4.6 con la quale abbiamo mosso

i primi passi nell'ambiente ABAP. Si può schematizzare l'evoluzione nella tabella riportata più sotto.

Il bello dell'ambiente SAP è che nonostante l'evoluzione tecnologica e funzionale, l'intero ERP rimane coerente con se stesso e i vecchi programmi continuano a "girare" con al limite pochi aggiustamenti da ricercare fra l'utilizzo di oggetti complessi che magari, nella nuova versione, hanno avuto in dono un parametro in più.

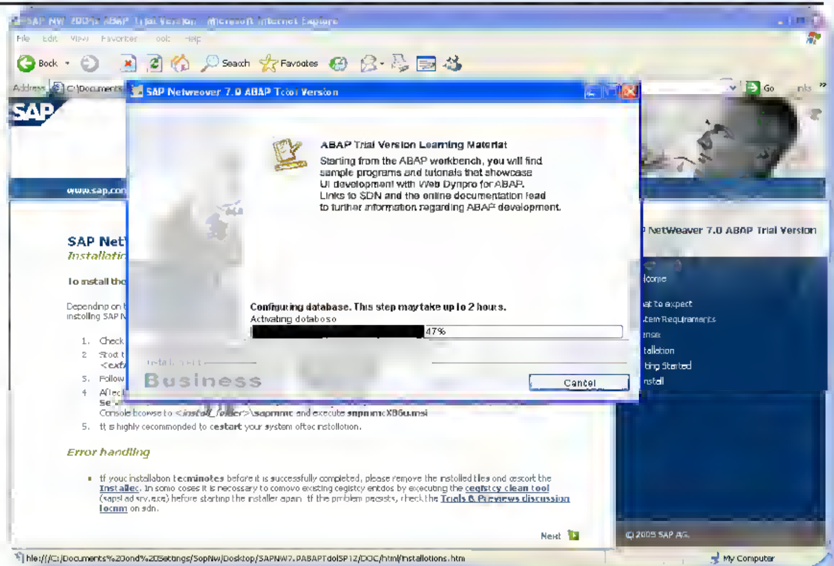
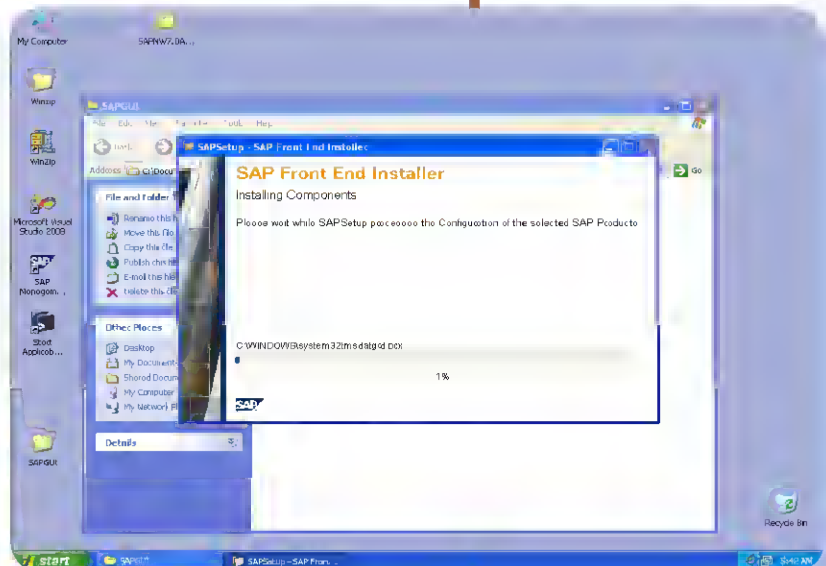


Figura 2.

SAP R/3 4.0	mySAP ERP 6.0	NetWeaver SOA
Client-server	Web Application Server	SOA Bus – Web services

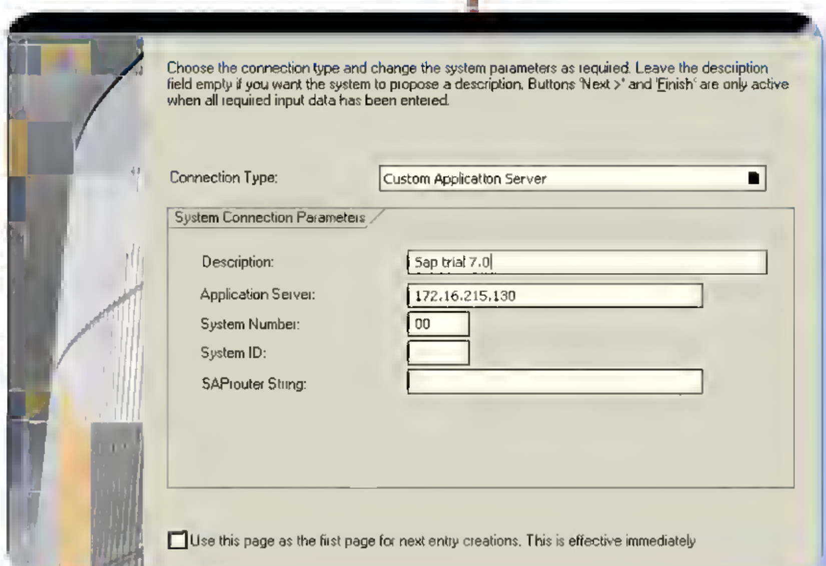
Figura 3.



Anche ABAP, il linguaggio di programmazione "storico" di SAP, è rimasto pressoché invariato mentre ad esso si è affiancato (già dalla versione 6) il più moderno (o "alla moda") Java. Come abbiamo già avuto modo di rilevare nelle puntate del corso pubblicate su Jurassic News, in effetti pochi usano Java e solo se i loro progetti necessitano di oggetti e librerie "esotiche" difficilmente riproducibili nell'ambiente ABAP.

L'installazione, rispettate le poche regole elencate nei requisiti di sistema, non presenta intoppi e prosegue senza difficoltà. L'unica "sorpresa", se così si può dire, è la constatazione che al termine del processo di setup, l'ambiente va ad occupare qualcosa più di venti giga. Da tenerne conto se si intende installare in una macchina virtuale. Un'altra idiosincrasia è il fatto che l'installazione della GUI, cioè la parte Client dell'ambiente (figura 4), deve essere lanciata dopo aver copiato la cor-

Figura 4.



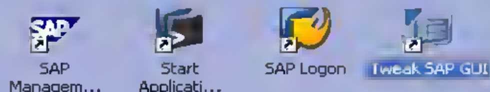


Figura 5.

rispondente directory dal DVD di installazione al file system (ad esempio sulla scrivania) del computer ospite. E' strana questa cosa e non ho mai capito se questo succede solo a me, visto che altri tecnici cui ho rivolto la domanda non mi hanno dato importanza, come se stessi dicendo una stupidaggine. Comunque sia, siete avvisati: se il setup della GUI vi da un errore di protezione, copiate la directory sul disco e partite da lì.

Assieme al server e alla GUI client, è possibile (e direi consigliato), caricare le utility di gestione del database (Figura 3).

Al termine ci si ritrova con quattro icone sul desktop (figura 5)

Si hanno a disposizione i seguenti programmi: SAP Management Console, Start Application Server, SAP Logon e Tweak SAP GUI.

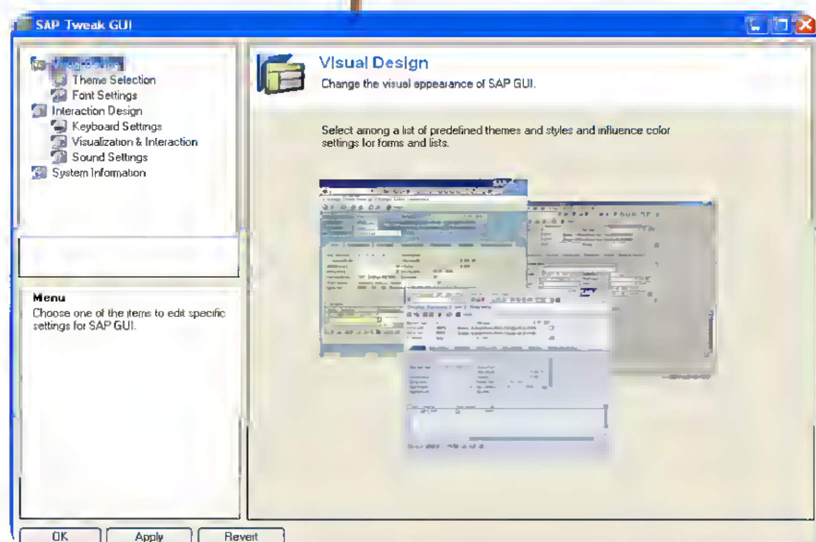
Per far partire il servizio si deve lanciare Start Application Server che si presenta come una console DOS (figura 7). Questa console può essere anche chiusa, contrariamente alle versioni precedenti dell'ambiente Trial che richiedevano di lasciarla in esecuzione.

A Questo punto va configurata la connessione al server (si tratta, ricordiamo di un ambiente client-server). Attraverso "SAP Logon" deve essere impostato l'indirizzo di rete e i parametri del server da raggiungere (figura 4)

Per configurare i parametri (colori, font,..) della GUI la SAP ha messo a disposizione un configuratore (Tweak SAP GUI) (figura 6)

Al momento del LOGIN al sistema si possono scegliere tre diversi utenti, differenziati per ruolo: SAP* è un utente amministratore, così come DDIC (ricoprono utilizzi diversi) e il "classico" BCUSER che è l'account di sviluppatore. Tutti e tre hanno come password "minisap" (Figura 7).

Figura 6.



Un programmatore SAP che provenga da una release precedente dell'ambiente workbench si ritrova famigliarmente con gli stessi strumenti, menù e tools delle precedenti versioni. In realtà esistono minime differenze, soprattutto esistono tool aggiuntivi che rendono ancora più completa la dotazione degli strumneti disponibili al coder.

Quello che ha subito un opportuno svecchiamento è l'editor (figura 8), adattato ora molto meglio alle moderne tecniche di codifica.

Scrittura, check e attivazione del codice si svolge esattamente nella forma consueta (Figura 9).

Client 000

User BCUSER

Password ************

Language EN

Information

SAP NetWeaver 7.0 ABAP Trial Version

Login information:

Client	User	Password
000	SAP*	minisap
000	DDIC	minisap
000	BCUSER	minisap

Figura 7.

Conclusione.

Un plauso alla SAP AG che continua con lo stesso entusiasmo e con la stessa caparbietà a fornire un ambiente educational/test per avvicinare sempre più persone al proprio sistema ERP. Certo la crisi economica sta colpendo anche questa azienda, che recentemente ha annunciato un piano di tagli preoccupante, ma in ogni caso rimane ben salda al top dei sistemi informativi aziendali nel mondo.

[Mx]

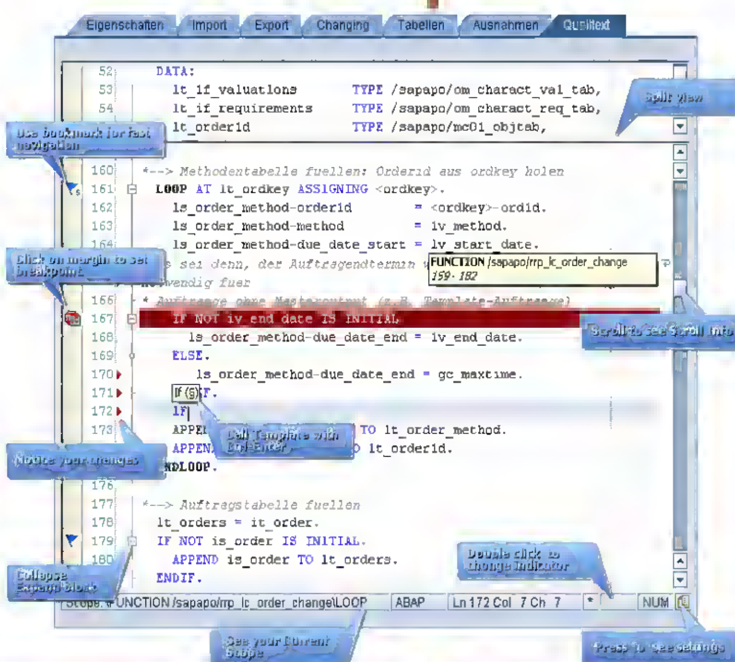
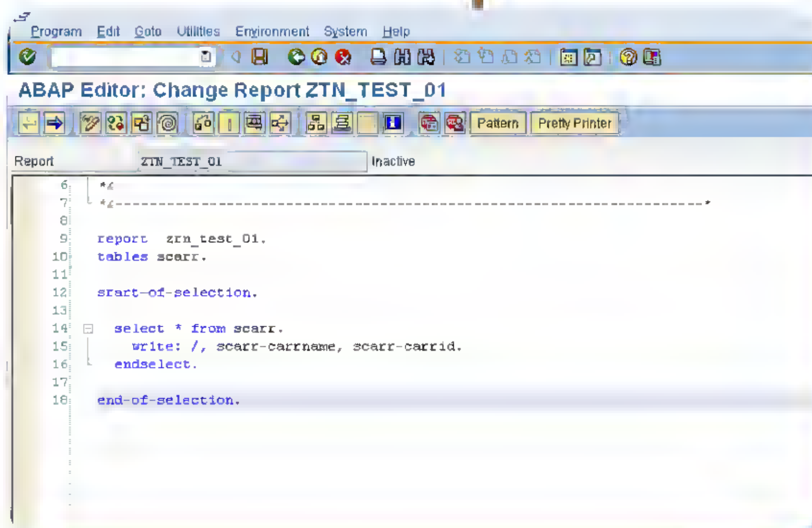


Figura 8.

Figura 9.



Nota della redazione

Sono pochissime le occasioni per sentire parlare di SAP dal punto di vista del programmatore in Italia. Abbiamo colto l'occasione di conoscenza di una persona che lavora nel campo come consulente free lance per curare prima il corso di ABAP (terminato da poco) ed ora per lanciare una vera rubrica fissa dedicata al sistema leader di mercato nell'informatica gestionale.

Apple Club

Tutti i linguaggi dell'Apple (12)

La mela come paradigma della programmazione

```
PL/BIT COMPILER V2.2
WRITTEN BY
CARLO MAGNAGHI

SOURCE TEXT
```

Il linguaggio PL in una particolare implementazione sponsorizzata dalla rivista Bit.

Il linguaggio PL/Bit

Domanda: -"Ma se il linguaggio che sto cercando non è disponibile per il mio calcolatore?"

-"Nessun problema: se manca un compilatore basta costruirselo!".

Certo che questa affermazione mette la parola fine alla ricerca della disponibilità del linguaggio del quale abbiamo proprio bisogno. Per contropartita è anche abbastanza inquietante, dal momento che chi conosce di programmazione sa benissimo che non è proprio una passeggiata costruirsi un interprete e peggio ancora un compilatore. Passi per un macro-assembler che tutto sommato è una traduzione uno a uno fra istruzione e codice binario, ma un "vero" linguaggio ad alto livello, magari strutturato?

Prima di tutto è naturale chiedersi che necessità ci sia di disporre proprio di quel particolare interprete o compilatore quando è noto a tutti che tutti i linguaggi formali sono equivalenti. Il che significa in parole povere che si può implementare un algoritmo in qualunque linguaggio di programmazione purché sufficientemente completo.

La risposta è che una macchina come l'Apple II è usata molto per il settore educational, cioè per quelle attività di insegnamento dell'informatica e della programmazione dei

calcolatori. Se disponiamo di una descrizione teorica di un linguaggio di programmazione, cosa che succede ad esempio sul libro di Nikolaus Wirth "Algorithms + Data structures = Programs", ci piacerebbe averne una implementazione pratica con la quale sperimentarne le caratteristiche.

Preso atto di questa considerazione e constatato che nel volume citato esiste la descrizione formale del linguaggio di programmazione chiamato PL/1, un certo Carlo Magnaghi ha deciso di costruirsene una implementazione che gira sull'Apple IIe sotto DOS. La descrizione del suo lavoro è apparsa su i fascicoli di programmazione allegati alla rivista Bit e che si chiamava "Bit Riservato Personal" nell'anno 1984 (precisamente sui fascicoli numerati 52, 53 e 54).

L'articolo è necessariamente molto lungo, comprendendo tutti i sorgenti del compilatore: un sorgente BASIC di circa 500 Statement più un listato assembler, oltre che tutte le spiegazioni del caso e tre sorgenti in PL/Bit come test ed esempio delle capacità del linguaggio e del compilatore in se e per se.

Davvero è da rendere omaggio all'autore perché è un lavoro accurato, per nulla semplice e di grande spessore tecnico!

Messe da parte le doverose lodi al merito, passiamo a descrivere il prodotto.

Partendo da un sorgente di testo che contiene il codice da compilare, il programma in Applesoft (esiste anche in versione compilata con TASC), crea il codice oggetto e produce infine l'eseguibile con l'ausilio di una libreria di funzioni standard a corredo del compilatore stesso.

Al termine della compilazione il file binario prodotto è eseguibile con il consueto comando BRUN del sistema operativo DOS 3.3.

Non ci sembra il caso di dilungarci nell'esame di come sia costruito questo compilatore che, lo ripetiamo, è un eccellente esempio di programmazione. Piuttosto

```
          PL/BIT COMPILER V2.2
              WRITTEN BY
            CARLO MAGNAGHI

SOURCE TEXT HILBERT
OBJECT TEXT HILBERT.O
OBJECT CODE STARTS AT 4000
STACK STARTS AT 8000
SHAPE TO LOAD

INTERFACE
```

Il compilatore in azione, dopo aver chiesto il nome del file sorgente e i parametri di compilazione

Il compilatore non produce direttamente l'eseguibile ma un sorgente assembler che deve poi a sua volta essere assemblato per generare il vero file binario pronto per l'esecuzione.

```
LIST WHAT TEXT FILE? HILBERT.O
TYPE:  CONTROL-S   TO STOP LISTING
       ANY KEY     TO CONTINUE

 STA TOS
 LDY #0
 LDA (136),Y
 STA LSD+0
 INY
 LDA (136),Y
 STA MSD+0
 INC TOS
 LDA #0
 STA MSD+1
 LDA #0
 STA LSD+1
 INC TOS
 JSR GT.
 LDA MSD
 ORA LSD
 BNE LBL.0
```

```
,(Y*3)/2;
  CALL B I-1;
 END
END;

BEGIN IT:=0;
REPEAT IT:=1-IT;
 CALL SHOWPAGE 1;CALL USEPAGE 1;
 CALL FILLSCREEN 0;CALL HCOLOR 3;
 I:=0;H:=H0;X0:=H/2;Y0:=X0;
 REPEAT
  IF IT=0 THEN CALL FILLSCREEN 0;
  I:=I+1;H:=H/2;
  X0:=X0+H/2;Y0:=Y0+H/2;
  X:=X0+28;Y:=Y0;CALL HPLOT (X*3)/2,(Y*3
)/2;
  CALL A I;
 UNTIL I=N
UNTIL 1=2
END.

***** 0 ERRORS FLAGGED

]※
```

Nessun errore di compilazione: bene!

facciamo un giro nel linguaggio PL e nella sua implementazione "casareccia" PL/Bit.

Purtroppo non possiamo allegare il PDF della serie di articoli, perché coperti da copyrights, per cui vi dovete accontentare di un sommario overview del sottoscritto sulle principali caratteristiche del linguaggio.

Partiamo con leggere un sorgente di esempio. Si tratta del programma Queen, abbastanza famoso e che consiste nel trovare le posizioni di otto regine sulla scacchiera in modo che non siano mutualmente minacciate.

Nel box listato 1 l'intero sorgente. Così a prima vista noterete che praticamente è Pascal, del resto il Wirth è proprio l'ideatore del linguaggio Pascal e questo PL/1 è evidentemente uno studio preliminare al linguaggio che poi l'ha reso popolare nel mondo.

L'assegnazione del valore ad una variabile tramite l'operatore ":=" direi che è proprio la firma di Wirth che era un purista ossessionato dalla semantica dei linguaggi formali.

Si parte con la dichiarazione delle variabili globali. Il linguaggio tratta solo variabili intere e quindi non c'è alcuna necessità di dichiararne il tipo. Quello che segue ARRAY sono dichiarazioni di vettori con i rispettivi limiti inferiore e superiore.

Un programma PL è composto da un corpo e da eventuali procedure che sono richiamate all'interno delle istruzioni. Il programma Queen comincia con l'istruzione BEGIN in fondo alla pagina e termina con END seguito dal punto. Il punto ".", come in Pascal, determina la fine del sorgente.

Le due subroutine, chiamate rispettivamente TRY e PRINT si chiamano reciprocamente e ricorsivamente, nel senso che TRY chiama la PRINT quando ha bisogno di mandare in output un risultato e la TRY chiama se stessa per iterare la ricerca delle soluzioni.

Una cosa strana e che salta immediatamente all'occhio è la presenza di istruzioni precedute dal simbolo chiocchiola "@" e che in pratica è un codice assembler in-line.

Ad esempio nella procedura PRINT serve per chiamare le routine di output sul video contenute nel firmware dell'Apple IIe. Mentre l'istruzione in-line

@ JSR $FD8B

```
VAR I;
ARRAY
        A I..8,
        B 2..I6,
        C -7..7,
        X I..8;

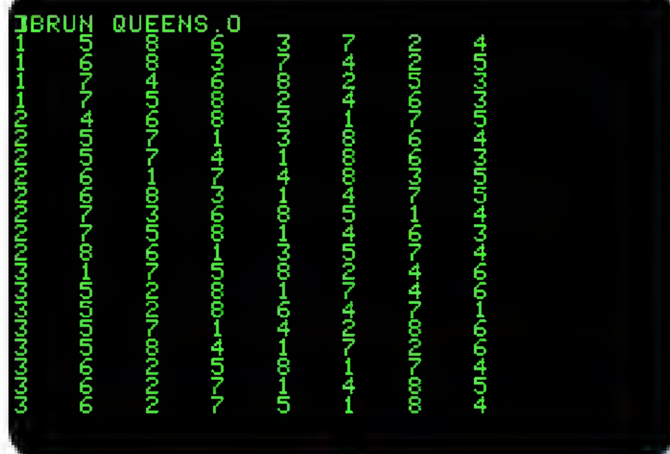
PROCEDURE PRINT;
VAR K,NUMOUT;
BEGIN
        K:=0;
        REPEAT
                K:=K+I;
                NUMOUT:=X(K);
                @ LDA LSD
                @ JSR $FDE3
                @ JSR $F948
        UNTIL K=8;
        @ JSR $FD8B
END;

PROCEDURE TRY I;
VAR J;
BEGIN
        J:=0;
        REPEAT
                J:=J+I;
                IF A(J) AND B(I+J) AND C(I-J) THEN
                        BEGIN
                                X(I):=J;
                                A(J):=0;
                                B(I+J):=0;
                                C(I-J):=0;
                                IF I<8 THEN
                                        CALL TRY I+I
                                ELSE
                                        CALL PRINT;
                                A(J):=I;
                                B(I+J):=I;
                                C(I-J):=I
                        END
        UNTIL J=8
END;

BEGIN
        I:=0;
        @ JSR $FD8B
        REPEAT
                I:=I+I;
                A(I):=I
        UNTIL I=8;

        CALL TRY I
END.
```

Listato 1. Il sorgente del programma Queen.



La macchina, bontà sua, sta sputando fuori le soluzioni del gioco delle otto regine. I numeri si riferiscono alla posizione della regina nelle rispettive righe. Ad esempio la prima soluzione: 1, 5, 8, 6, 3, 7, 2, 4 significa che nella prima colonna la regina è nella casella 1, nella seconda colonna c'è una regina nella posizione 5 e così via.

che si trova nel corpo del sorgente principale, serve per pulire il video.

La strada seguita dall'autore del compilatore è stata quella di focalizzarsi sulla traduzione delle istruzioni proprie del linguaggio usando quello che è disponibile per l'I/O già implementato nella macchina.

Un pochino più articolato il sorgente HANOI (box listato 2) che risolve la famosa torre di Hanoi, altro tormentone da studenti di informatica primo anno.

In questo caso si parte con la dichiarazione INTERFACE che in pratica dichiara cosa vogliamo usare come output e, lo si capisce dalle direttive che seguono la dichiarazione, stiamo lavorando con la pagina semigrafica 1 dell'Apple.

All'interno del sorgente notate le macro come CALL HPLOT, HPLOTTO, etc... di chiaro significato. In questo sorgente c'è anche la dichiarazione di costante (statement CONST) dichiarata all'inizio prima delle variabili. La sequenza delle dichiarazioni è rigida: prima le costanti, poi le variabili semplici e poi gli array. Del resto si tratta di limitazioni più che accettabili.

Vediamo come le procedure possono ricevere parametri come nella MOVEDISK che riceve due coordinate: ORG e DST, evidentemente con significato di origine e destinazione. Ricordiamo che il "gioco" consiste nello spostare dei dischi di dimensione variabile da un piolo iniziale al piolo finale utilizzando al più un piolo intermedio con il vincolo di non appoggiare mai un disco su uno di diametro inferiore.

Usare il compilatore

Sul dischetto distribuito con il compilatore esistono due versioni: il sorgente Applesoft classico e il compilato via TASC che richiede però la presenza del RUNTIME per essere eseguito.

Devo dire che la differenza di esecuzione è abbastanza limitata fra le due versioni, almeno sui sorgenti corti come quelli di esempio. Se si vuole usare l'interprete si lancia con RUN PL/BIT, se si desidera usare il compilato si può usare il file di comandi che risiede sul dischetto con nome COMPILER e va fatto partire con

EXEC COMPILER

questo carica la libreria TASC e poi l'eseguibile che si chiama PL.OBJ

In ogni caso il compilatore chiede il nome del file sorgente, il nome dell'oggetto da produrre, l'indirizzo di memoria (in decimale) dove si

```
INTERFACE
USEPAGE I,
SHOWPAGE I,
FILLSCREEN I,
HCOLOR I,
HPLOT 2,
HPLOTTO 2;

CONST        YMOVE=I40;

VAR          NUM,I,J;
ARRAY        XMID I..20,
             WIDTH I..20,
             YLOW I..20;

PROCEDURE MOVEDISK ORG,DST;
VAR XO,YO,XD,YD,XI,X2,XM,I,DISK,D;

BEGIN
        XO:=ORG*90+50;
        XD:=DST*90+50;
        YO:=200;
        YD:=I90;
        I:=0;
        REPEAT
                I:=I+I;
                IF YLOW(I)<YO AND XMID(I)=XO THEN
                BEGIN
                        YO:=YLOW(I);
                        DISK:=I
                END;
                IF YLOW(I)<YD AND XMID(I)=XD THEN
                        YD:=YLOW(I)
        UNTIL I=NUM;
        YD:=YD-5;
        XM:=XMID(DISK);
        XI:=XM-WIDTH(DISK);
        X2:=XM+WIDTH(DISK);
        REPEAT
                CALL HCOLOR 0;
                CALL HPLOT XI,YO;
                CALL HPLOTTO X2,YO;
                CALL HCOLOR 3;
                CALL HPLOT XI,YO-4;
                CALL HPLOTTO X2,YO-4;
                YO:=YO-I
        UNTIL YO<YMOVE;
        D:=2*(DST>ORG)-I;
....
```

Listato 2.
Il sorgente del programma Hanoi.

```
        PL/BIT COMPILER V2.2
             WRITTEN BY
           CARLO MAGNAGHI



SOURCE TEXT HILBERT
OBJECT TEXT HILBERT.O
OBJECT CODE STARTS AT 4000
STACK STARTS AT 8000
SHAPE TO LOAD

INTERFACE
```

Si comincia la compilazione del programma Hilbert.

vuole parta il codice e l'indirizzo dello stack (viene consigliato per lo stack il valore 2048).

Durante la compilazione, che avviene in un'unica passata, il compilatore stampa le righe di sorgente che sono di volta in volta tradotte. Sembrerebbe una perdita di tempo e nessun compilatore oggi si sognerebbe mai di proporre questa opzione, ma invece per le macchine "antiche" questo era utile: si capiva che il lavoro andava avanti e si identificava immediatamente la riga con l'eventuale errore bloccante.

Per quanto riguarda la velocità di traduzione da sorgente a codice oggetto non è nulla di paragonabile ad un Turbo Pascal, tanto per fare confronti non troppo distanti.

Prima di tutto già la preparazione dell'esecuzione (caricamento libreria TASC e oggetto PL.OBJ), impiega buoni quaranta secondi prima di presentare la domanda del nome del sorgente da compilare.

Un sorgente come HILBERT di circa un centinaio di statement, impiega due minuti e trentacinque secondi su un Apple IIe a 1 MHz, per arrivare alla fine.

Che volete farci: si lancia e si va a prendere il caffé, un classico!

Il codice sorgente così ottenuto consiste in un file di testo con l'elenco delle istruzioni mnemoniche sono da assemblare con un assemblatore standard (non presente sul dischetto del compilatore).

Assemblare il tutto

Noi abbiamo deciso di usare l'assemblatore EDASM che è presente sul dischetto "Utility Disk" assieme al DOS 3.3 quando si compra un Apple. Per prima cosa prepariamo un dischetto di lavoro che contenga tutto quello che serve. Noi lo abbiamo fatto partendo da un floppy vuoto e caricandoci tutti i programmi indispensabili per l'intera sessione.

Per quanto riguarda il compilatore PL/BIT, ammesso di voler usare la versione non compilata, serve solamente il programma Applesoft PL/BIT.

Abbiamo messo sul floppy anche il programma GET TEXT che serve per farsi stampare a video un file di testo, il programma EDITOR (sempre trovato sul dischetto utility del DOS) per editare i sorgenti.

Per l'assemblatore serve il programma EDASM, la sua parte compilata che si chiama ASM.OBJ e un file del quale non abbiamo capito lo scopo ma che se non è

presente manda in dump l'assemblatore: ASMIDSTAMP.

Serve inoltre la libreria PL chiamata RUNTIME che viene assemblata assieme al file oggetto per produrre il compilato eseguibile.

La fase di assemblaggio prevede di invocare il programma assembler:

RUN EDASM

e quindi al prompt dello stesso (il carattere ":" due punti) digitare il comando:

ASM SIER

(Nota: SIER è il nome del file object ottenuto dalla compilazione del sorgente Sierpinsky).

L'assemblatore produce sul dischetto il file con nome SIER.OBJ0 che è appunto l'eseguibile. L'esecuzione del programma compilato non richiede runtime e si ottiene con il semplice comando BRUN:

BRUN SIER.OBJ0

```
(C) COPYRIGHT 1979
APPLE COMPUTER INC

:ASM SIER

PRESS ANY KEY TO CONTINUE
SOURCE FILE: SIER..........
```

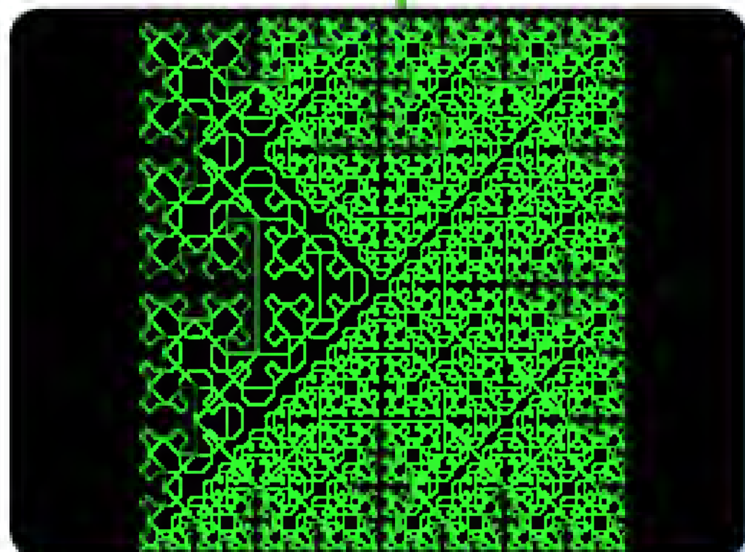
Qui è in funzione l'assembler

Conclusione

Aldilà dell'effettiva utilità pratica del compilatore PL/Bit, non possiamo che constatare con non poca soddisfazione come il sistema della Apple si presentava adatto alla stesura di codice anche di media complessità. Giocava sicuramente a favore una conoscenza "intima" della macchina e delle sue potenzialità che era disponibile a coloro che si potevano dedicare anima e corpo e soprattutto che avevano tempo per esplorarne le potenzialità.

[Sm]

Data la scarsa reperibilità di questo software abbiamo deciso di allegare il dischetto a questo numero della rivista. Prima di fare ciò abbiamo cercato inutilmente di rintracciare il sigron Magnaghi per chiedere il permesso. Purtroppo non ci siamo riusciti, così confidiamo di non fargli cosa sgradita permettendo a tanti appasssionati della Mela ai quali magari a suo tempo era sfuggito il suo lavoro di farlo rivivere. Potete scaricare l'immagine del floppy da 140 Kb formato DSK dal sito della rivista.



Ed ecco il programma Sierpinsky in esecuzione.

Jurassic News Anteprima

Sul prossimo numero ci aspettano:



Il racconto

My story (4) - Il nuovo lavoro

L'intervista

Intervista ad A. Zucch

Retro Linguaggi

LISP (parte 3)

DIVERTIRSI CON IL CALCOLATORE

giochi, simulazioni e grafica

a cura di VIRGINIO SALA

TAMC

Algoritmi di SORT (parte 6)

RETROGAMING TIMES monthly

'70s '80s '90s

COVERING 3 DECADES OF GAMING

Table Of Contents